CHAPTER 2

LITERATURE SURVEY

Association rule mining has been an active research area in data mining, for which many algorithms have been developed. (Srikant et al 1997; Garofalakis et al 1999; Zaki et al 2000; Zaki et al 2001; Salleb et al (2007)). Before the review of relevant existing sampling based and incremental mining algorithms, the ways for improving the association rule mining techniques are presented.

2.1 IMPROVING THE EFFICIENCY OF ASSOCIATION RULE MINING

Discovering frequent patterns from large datasets using ARM algorithms has been extensively employed in the literature (Goethals, 2003; Sotiris and Dimitris, 2006) due to great capabilities of ARM algorithms in improving business profit. Over the past decade, a variety of research efforts improved the processing time of ARM algorithms like the one introduced by Ceglar and Roddick (2006). Goethals et al (2005) have reduced the output set size that is generated by ARM. Overviews of methods and techniques for improving the efficiency of ARM are presented below.

2.1.1 Sampling

Sampling is a powerful data reduction technique that has been applied to a variety of data mining algorithms for reducing computational overhead. In the context of association rules, sampling can be utilized to gather quick preliminary rules. This may help the user to direct the data mining process by refining the
criterion for “interesting” rules. Sampling can speed up the mining process by more than an order of magnitude by reducing I/O costs and drastically shrinking the number of transaction to be considered. The validity of the sample is determined by two characteristics namely the size and quality of the sample.

The quality, in the context of statistical sampling techniques, refers to whether the sample captures the characteristics of the database. The highest quality sample would be an exact miniature of the database; it would preserve the distributions of individual variables and the relationships among variables (Tsau, 2004). The quality of the sample for association rule mining can be improved by considering factors like transaction length and transaction frequency (Basel et al, 2009). A number of studies were conducted to propose efficient methods for mining association rules by reducing either the CPU computation time or the disk access overhead. Some studies considered the usage of sampling techniques for reducing the processing overhead. Most of the prior works on sampling have concentrated on speeding up the phase by running a frequent itemset mining algorithm only on a small sample of the database.

2.1.2 Reducing the Number of Passes

The disadvantage of Apriori algorithm made the researchers to think about new techniques to mine frequent patterns. The 2 main negative sides are the possible need of generating a huge number of candidates if the number of frequent 1-itemsets is high or if the size of the frequent pattern is big, the database has to be scanned twice repeatedly to match the candidates and determine the support. Mining the frequent patterns without candidate generation would be a big improvement over Apriori. That is what the frequent pattern growth (FP-growth) algorithm does (Han et al, 2000).
Wang et al (2002) presented PRICES, an efficient algorithm for mining association rules. Their approach reduces large itemset generation time, which dictates most of the time in generating candidates by scanning the database only once. Another algorithm called Matrix Algorithm developed by Yuan and Huang (2005) generates a matrix which entries 1 or 0 by passing over the cruel database only once. The frequent candidate sets are then obtained from the resulting matrix. Association rules are then mined from the frequent candidate sets.

### 2.1.3 Hash-Based Itemset Counting

A hash technique is very efficient in generating the candidate item sets, in particular for the large two-itemsets, thus greatly improving the performance bottleneck of the entire process.

Soo et al (1997) proposed Direct Hashing and Pruning [DHP] algorithm, an effective hash based technique for mining the association rules. This algorithm employs effective pruning techniques to progressively reduce the transaction database size. DHP utilizes a hashing technique to filter the ineffective candidate frequent 2 itemsets. DHP also avoids database scans in some passes as to reduce the disk I/O cost involved.

Another novel hash-based approach for mining frequent itemsets over data streams was developed by En et al (2011). The algorithm compresses the information of all itemsets into a structure with a fixed hash-based technique. This approach skillfully summarizes the information of the whole data stream by using a hash table to estimate the support counts of the non-frequent itemsets and keeps only the frequent itemsets for speeding up the mining process.

Another algorithm Inverted Hashing and Pruning (IHP) is implemented by John and Soon (2000) for mining association rules between words in text
databases. The characteristics of text databases are quite different from those of retail transaction databases and existing mining algorithms cannot handle text databases efficiently because of the large number of itemsets (i.e., words) that need to be counted. Two well-known mining algorithms, the Apriori algorithm (Agarwal et al, 1997) and Direct Hashing and Pruning (DHP) algorithm (Soo et al, 1997) are evaluated in the context of mining text databases and are compared with the proposed IHP algorithm. It has been shown that the IHP algorithm has better performance for large text databases.

### 2.1.4 Transaction Reduction

Transaction reduction is another way that helps in mining association rules effectively. It relies on a concept that a transaction that does not contain any frequent k-itemset is useless in subsequent scans.

AprioriTid algorithm (Rakesh and Ramakrishnan, 1994) is another way of improving the performance of Association Rules. This algorithm is used to construct the frequent itemset. The main idea of all these algorithms is according to the theory that the subset of a frequent itemset is a frequent itemset and the superset of an infrequent itemset is an infrequent itemset. They scan the database repeatedly to mine the association rules. Another feature for algorithm AprioriTID is that the support of the candidate frequent itemsets are calculated only at the first time it scanned the database D and also generated candidate transaction database D’ which only includes the candidate frequent itemsets. Then the latter mining is based on the database D’. It reduces the time of I/O operation because D’ is smaller than D.
Another approach called MTR-FMA (modified transaction reduction based frequent itemset mining algorithm) proposed by Thevar and Krishnamoorthy (2008) maintains its performance even at relative low supports.

2.1.5 Partitioning

Various approaches to generate large item sets have been proposed based on partitioning the set of transactions. In this case, D is divided into p partitions $D_1, D_2,...,D_p$. Partitioning may improve the performance of finding large item sets in several ways. By using partitioning, parallel and/or distributed algorithms can be easily created, where each partition could be handled by a separate machine.

Cheung et al (1996) 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. Distributed Pruning is then done. Cheung et al (1998) proposed FPM (Fast Parallel Mining) for Association rule mining has been proposed. It adopts Count Distribution approach and has incorporated two powerful candidate pruning techniques. It has a simple communication scheme which performs only one round of message exchange in each iteration. Parthasarathy et al (2001) have presented an excellent survey on parallel association rule mining with shared memory architecture covering most of challenges and approaches adopted for parallel data mining.

2.1.6 Adding Extra Constraints

Another type of association rule mining involves in retrieving patterns by adding extra constraints on the structure of patterns. Techniques applicable to constraint-driven pattern discovery can be classified into the following groups (Sotiris and Dimitris, 2006):
- Post-processing (filtering out patterns that do not satisfy user-specified constraints after the actual discovery process).
- Pattern filtering (integration of pattern constraints into the actual mining process in order to generate only patterns satisfy pattern constraints).
- Dataset filtering (restricting the source data set to objects that can possibly contain patterns that satisfy pattern constraints).

Wojciechowski and Zakrzewiez (2002) proposed a constraint based algorithm that improves the efficiency of constraint based frequent pattern mining by using dataset filtering techniques. Dataset filtering conceptually transforms a given data mining task into an equivalent one operating on a smaller dataset. Rapid Association Rule mining (RARM) presented by Das et al (2001) is another method that uses a tree structure to represent original database and avoids candidate generation process. Constraints were applied during the mining process to generate only those association rules that are interesting to the users which guarantees the improvement of the efficiency of the existing mining algorithm.

Tien et al (2003) presented a category based algorithm as well as the associated algorithm for constraint rule mining based on Apriori. This approach reduces computational complexity of mining process by passing most of the subsets of final itemsets.

2.1.7 Association Rule Clustering System

Association Rule Clustering is useful when the user desires to segment the data. Lent et al (1997) proposed a Clustering Association rule in which they measure the quality of the segmentation generated by ARCS (Association Rule Clustering System). It uses the minimum description length (MDL) principle of encoding the clusters on several databases including noise and errors. Scale-up
experiments show that ARCS using the BitOp algorithm scales linearly with the amount of data.

Pi and Qin (2008) proposed a new Fuzzy Clustering Algorithm on Association Rules for Knowledge Management. A fuzzy simulation degree and simulated matrix for association rules are put forward and a new algorithm based on dynamic tree is used for implementing the fuzzy clustering. The experimental results show that this algorithm clusters the Association rules efficiently.

Rajendra and Dev (2009) recently proposed a cluster based algorithm that uses a novel approach to the insignificant transactions dynamically. During a particular pass only those clusters that seem to be statistically useful are scanned and as a consequence all insignificant tuples are filtered dynamically. The results of the algorithm show that removing false frequent items and insignificant transactions dynamically improves the performance of association rule mining.

2.2 SAMPLING BASED ASSOCIATION RULE MINING ALGORITHMS

Several researches are available in the literature for sampling based association rule mining algorithms and Incremental mining algorithms are reviewed. In this section, a brief review of some of the significant researches is presented here.

Sampling approaches offer strong solutions for the problems of ARM algorithms to extract frequent patterns from large datasets. The concept of sampling algorithms is to apply the mining process on a sample dataset, instead of applying it on the entire dataset, thus reducing the processing time and memory space requirements.
The simplest way to apply sampling is to select a random sample. Using this random sample as an input for ARM has been analysed by (Zaki et al 1996). This method used a sequential random selection transaction without replacement from the original dataset. The transactions are selected in the same order as they appear in the original dataset. Chernoff bound is employed to provide the information of how close the actual occurrence of an itemset was in the sample compared to the expected count in sample. Based on itemset frequencies and using chernoff bounds, one can determine the sample size. They have the following contributions: i. Sampling can reduce i/o cost by drastically shrinking the number of transactions to be considered and ii. Sampling can provide greater accuracy with respect to the association rules. They have shown that sampling can speed up the mining process by more than an order of magnitude.

Toivonen et al (1997) presented an Association rule mining algorithm using sampling technique. The approach can be divided into 2 phases. In phase I, a sample of the database and all associations in the sample of the database are obtained. 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. Since the approach is dependent (probabilistic) on the sample containing all relevant associations, not all the rules may be found in the first pass. These associations that were deemed not frequent in the sample but were actually frequent in the entire dataset are used to construct the complete set of associations in phase 2.

Parthasarathy (2002) proposed an efficient method to progressively sample the dataset for association rules. This approach relies on a novel measure of model accuracy (self-similarity of associations across progressive samples). The identification of a representative class of frequent item sets that mimic (extremely
accurate) the self-similarity values across the entire set of association. This is an efficient sampling methodology that hides the overhead of obtaining progressive samples by overlapping it with useful computation.

Chen et al (2002) presented a novel two phase sampling algorithm for discovering association rules in large databases. These algorithm has 2 phases. In phase I, a large initial sample of transactions is collected and used to quickly and accurately estimate the support of each individual item in the database. In phase II, these estimated supports are used to either trim “outlier” transactions or select "representative" transactions from the initial sample, thereby forming a small final sample that more accurately reflects the statistical characteristics (i.e. itemset supports) of the entire databases. The expensive operation of discovering association rules is then performed on the final sample.

Bronnimann et al (2003) explored another sampling algorithm called epsilon approximation sample enabled (EASE). EASE uses Epsilon approximation methods to obtain the final sub sample by the process of repeated halving. This algorithm can process transactions on the fly, i.e. a transaction needs to be examined only once to determine whether it belongs to the final sub sample.

Surong et al (2005) presented another algorithm called EASIER which is an extension to EASE that was proposed by Bronnimann et al (2003) in two ways. 1) EASE is a halving algorithm i.e. to achieve the required sample ratio it starts from a suitable initial large sample and iteratively halves. EASIER on the other hand, does away with the repeated halving by directly obtaining the required sample ratio in one iteration. 2) EASE was shown to work on IBM Quest dataset which is a categorical count data whereas EASIER in addition to count data it was also shown on Continuous data such as Color Structure Descriptor (CSD) of images.
Chuang et al (2005) presented another progressive algorithm called Sampling Error Estimation (SEE) which aims to identify an appropriate sample size for mining association rules. SEE has two advantages. 1. SEE is highly efficient because an appropriate sample size can be determined without the need of executing association rules. 2. The identified sample size of SEE is very accurate (i.e.) the association rules can be highly efficiently executed on a sample of this size to obtain a sufficiently accurate result.

Venkatesan et al (2009) proposed a different view of analyzing the quality of solution by theoretical framework. Their contributions are twofold. First, the notions of e-close frequent item set mining and e-close association rule mining that help assess the quality of solutions obtained by sampling. Secondly, the frequent itemset mining and association rule mining problem can be solved satisfactorily with a sample size that is independent of both the number of transactions size and number of items. It has also been established that it is possible to speed up the entire mining process of association rule mining for massive databases by working with a small sample size while retaining any desired degree of accuracy. Their work also gives a comprehensive explanation for well-known empirical success of sampling for association rule mining.

Basel et al (2009) recently presented a parameterized sampling algorithm for association rule mining. This algorithm extracts sample datasets based on three parameters: transaction frequency, transaction length and transaction frequency length and it empirically shown that it achieves 95% accuracy which outperform two-phase algorithm that was proposed by Chen et al (2002).

Shi et al (2009) presented a sampling algorithm for mining association rules in distributed database (SMA) that adopts two-way sampling technique in every website and amalgamates those local samples into a sampling dataset.
In a empirical study, SMA was able to accelerate the speed and improve the accuracy of mining association rules in distributed database.

Wontae et al (2011) presented a new algorithm called IFAST that uses two phase sampling proposed by Chen et al (2002) algorithm for shortening the execution time at the cost of precision of the mining result. Previous FAST proposed by Chen et al (2002) algorithm has the weakness in that it only considered the frequent 1-itemsets in trimming/growing phase. Thus it did not have ways of considering multi-item sets including 2 itemsets. IFAST algorithm reflects the multi-item sets in sampling transactions. It improves the mining results by adjusting the counts of both missing item sets and false itemsets.

In most sampling based association rule mining algorithms, the samples are selected randomly from the large database without considering any of the characteristics of the database. Moreover, it has been highly difficult to determine an optimal sample size for effectively mining association rules.

In certain situations, the data miner has to perform sampling on the dataset before applying any algorithm. The main reason behind this is being too many data to mine. In such a case, a possible technique is random sampling. If classes are uniformly distributed, one may use random sampling before supervised learning.

If random sampling is used before an association rule algorithm, it may end up finding no rule. The reason is that association rule mining analyses the data as transactions. The issue with random sampling is that it will not take into account the continuous sequence of events. In the case of association rules, one should take a continuous subset of the data in order to get meaningful rules.
2.3 INCREMENTAL RULE MINING ALGORITHMS

This section discusses about the various approaches for incremental mining of association rules. In recent times, developing approaches for incremental mining of association rules has gained huge importance in real life applications. In the real world where large amounts of data grow steadily, some old association rules can become obsolete and new databases may give rise to some implicitly valid patterns or rules. Hence, updating rules or patterns is important. A simple method for solving the updating problem is to reapply the mining to the entire database, but this approach is time consuming. Several approaches to incremental mining have been proposed (Ayan et al, 1999; Cheung et al, 1997; Hong et al, 2001; Lee et al, 2001; Ng and Lam, 1999; Ng et al, 2001; Sarda and Srinivas, 1998; Thomas et al, 1997, Velosa et al, 2000; Sarasere, 1995) in the literature.

A brief review of some recent researches related to incremental mining of association rules is presented here.

Fast update (FUP) proposed by Cheung et al (1996) is one approach to association rules that can handle incremental update to reduce the size of the candidate to be searched in original large databases. In the real world where large amounts of data grow steadily, some old association rules can become useless and new databases may give rise to some implicitly valid patterns or rules. Hence, updating rules or patterns is important. The FUP algorithm is well known in relation to this issue. A simple method for solving the updating problem is to reapply the mining to the entire database. But this approach is time consuming. The FUP algorithm uses information from old frequent itemsets to improve its performance.
A temporal windowing technique for incremental maintenance of association rules is proposed by Chris et al (1997). The approach is based on the premise that the transactions outside a user defined time window are too old to contribute towards association rules of current interest. They also define a strong support threshold and near strong threshold levels for mining strong and near strong association rules. These near strong rules have the potential to become strong association rules during next time window. Consequently, their update algorithm retires old and outdated transactions and carries out mining using incremental database only.

Ling et al (2004) have proposed a general incremental updating method that can be used to alter the determined frequent item sets in case of inclusion, removal, and alteration of transactions. Based on adjusting FP-tree structures, they have designed an efficient algorithm called AFPIM (Adjusting FP-tree for Incremental Mining). The compact information of transactions involving frequent and pre-frequent items in the original database has been stored in their approach using a FP-tree structure. In most cases, by adjusting the FP-tree of the original database according to the altered transactions the FP-tree structure of the updated database could be obtained without the necessity of rescanning the original database. Experimental results have shown that the execution time of AFPIM is superior to that of other existing algorithms.

Ya and Yen (2006) has proposed a MIS-tree which is a FP-tree-like structure to store the imperative information regarding frequent patterns. Accordingly, for mining all frequent item sets, CFP-growth algorithm uses an efficient MIS-tree-based algorithm. It has been very difficult for users to set the appropriate thresholds for all items at a time, since each item can have its own minimum support. The support of items necessitates tuning by users and repeated
execution of the mining algorithm until satisfactory result is obtained. So, to reduce the time consumed by the tuning process they have also proposed an efficient algorithm which does not necessitate rescanning the database to maintain the MIS-tree structure. The superiority of their algorithm over the previous algorithm in terms of efficiency and scalability has been proved by the experiments performed on both synthetic and real-life datasets.

Chun et al (2008) have proposed the structure of pre-large trees for pre-large itemset concept based incremental mining of association rules. Rescanning the original database has not been necessitated by their proposed approach because of the properties of the pre-large concepts, until a number of new transactions are inserted. The superior performance of their proposed approach in incremental handling of new transactions has been demonstrated by experimental results.

Tzung et al (2008) have simplified the tree update process in the fast updated FP-tree (FUFP-tree) structure. Incremental FUFP-tree maintenance algorithm reduces the execution time for reconstructing the tree when new transactions are inserted. Experimental results have also shown that in addition to creating a tree structure similar to that created by the FP-tree algorithm, new transactions are handled faster by the proposed FUFP-tree maintenance algorithm. An excellent trade-off between execution time and tree complexity has been achieved by their proposed approach.

Muhaimenul et al (2008) proposed an extension of the FP-tree concepts and manipulation process to address the incremental update problem. Different types of changes such as additions, deletions and alterations have been handled successfully in their proposed approach. They have accomplished the target by creating and incrementally dealing with the entire FP-tree i.e., with one minimum
support threshold. Freedom of mining for lower minimum support values without the necessity of reconstructing the tree has been the other advantage of creating the entire FP-tree. However, the basic FP-tree structure may become invalid when the changes are directly reflected onto the FP-tree. Thus, to validate and to maintain the modified tree, a series of shuffling and merging operations are applied. The benefits of the proposed incremental approach over creating the FP-tree from scratch has been clearly highlighted by the experiments conducted on both synthetic and real datasets.

Syed et al (2008) presented a tree structure, called CP-tree (compact pattern tree), to provide the same mining performance similar to FP-growth method (restructuring phase) by capturing the database information with one scan (insertion phase). The CP-tree has created a highly compact frequency-descending tree structure at runtime by introducing the dynamic tree restructuring concept. They have also proposed an efficient restructuring method to restructure a prefix-tree branch-by-branch, called the branch sorting method. In addition, complete functionality for interactive and incremental mining has been provided by the CP-tree. The CP-tree has been shown to be efficient for frequent pattern mining, interactive and incremental mining, with a single database scan by extensive experimental results.

Hong and Wang (2010) have presented a modification algorithm to minimize the computation time required for maintaining the association rules when the records in a database are altered. They have minimized the need for rescanning original databases and saved the maintenance costs by utilizing the concept of pre-large item sets. Until a specified number of records have been modified, the proposed algorithm has not necessitated rescanning of original databases. The number of modified records permitted has been large for large
databases. This characteristic has been valuable, in particular for real-world applications

Tzung and Ching (2010) have simplified the tree update process in the fast updated FP-tree (FUFP-tree) structure. An incremental FUFP-tree maintenance algorithm reduces the execution time for reconstructing the tree when new transactions are inserted. Experimental results have also shown that in addition to creating a tree structure similar to that created by the FP-tree algorithm, new transactions are handled faster by the proposed FUFP-tree maintenance algorithm when compared to the batch FP-tree construction algorithm. An excellent trade-off between execution time and tree complexity has been achieved by their proposed approach.

Thus, association rules represent an important class of knowledge that can be discovered from data warehouses. As the databases grow, the discovered rules need to be verified and new rules need to be added to the knowledge base. Since mining afresh every time the database grows is inefficient, algorithms for incremental mining are still in progress. Hence, developing an incremental algorithm is critically important for maintaining the mined association rules when the database grows. The primary aim is to concentrate in avoiding or minimizing scans of the older database by using the intermediate data constructed during the earlier mining.