Information Technology Journal 9 (7): 1278-1293, 2010
ISSN 1812-5638

A Survey on Frequent Pattern Mining: Current Status and Challenging Issues

[1]A. Tiwari, [1]R.K. Gupta and [2]D.P. Agrawal
[1]Department of CSE and IT, MITS, Gwalior, India
[2]Union Public Service Commission, New Delhi, India

Abstract: Discovering association rules in huge databases is a core topic of data mining. This survey study aims at giving an overview of the previous researches done in this field, evaluating the current status of the work done and envisioning gaps in the current knowledge. The problem of mining association rules can be generalized in to two steps: (1) Finding all frequent itemsets and (2) generating rules from these itemsets. The first sub-task, which is to determine the frequent itemsets, is computationally expensive process. Counting the occurrences of itemsets requires a considerable amount of processing time. As a consequence, number of algorithms are proposed in literature for mining the frequent itemsets. Present study reviews frequent pattern mining algorithms and other related issues available in the literature.

Key words: Association rule mining, algorithms, data mining, frequent patterns, research issues

INTRODUCTION

Frequent pattern mining is an essential step in the process of association rule mining and has been a focused theme in data mining research for over a decade. Abundant literature has been dedicated to this study and tremendous progress has been made, ranging from efficient and scalable algorithms for frequent itemset mining in transaction databases to numerous research frontiers, such as sequential pattern mining, structured pattern mining, correlation mining, associative classification and frequent pattern-based clustering, as well as their broad applications. Following sections present an overview of the current status of frequent pattern mining. However, there are still some challenging research issues that need to be explored.

Frequent patterns are itemsets, subsequences, or substructures that appear in a data set with frequency no less than a user-specified threshold. For example, a set of items, such as milk and bread, that appear frequently together in a transaction data set, is a frequent itemset. A subsequence, such as buying first a PC, then a digital camera and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern. A substructure can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences. If a substructure occurs frequently in a graph database, it is called a (frequent) structural pattern.

Finding frequent patterns plays an essential role in mining associations, correlations and many other interesting relationships among data. Moreover, it helps in data indexing, classification, clustering and other data mining tasks as well. Thus, frequent pattern mining has become an important data mining task and a focused theme in database community.

Frequent pattern mining was first proposed by Agrawal *et al*. (1993) for market basket analysis in the form of association rule mining. It analyses customer buying behavior by finding associations between the different items that customers place in their shopping baskets.

Researchers have proposed several improved algorithms for generating frequent itemsets. These algorithms differ in their ways of traversing the itemset lattice and the ways in which they use the anti-monotone property of itemset support. Another dimension where the algorithms differ is the way in which they handle the database; i.e., how many passes they make over the entire database and how they reduce the size of the processed database in each pass. On the basis of these issues, next subsection describes a representative set of the major algorithms proposed.

FREQUENT PATTERN MINING METHODOLOGIES

The very first algorithm was AIS (Agrawal *et al*., 1993). It was proposed to address the problem of association rule mining. This is a multi-pass algorithm in which candidate itemsets are generated while scanning the database by extending known-frequent itemsets with items from each transaction. An estimate of the supports of these candidates is used to guide whether these candidates need to be extended further to produce more

Corresponding Author: Akhilesh Tiwari, Department of CSE and IT, MITS, Gwalior, India candidates. The main problem of the AIS algorithm is that it generates too many candidates that later turn out to be infrequent. Another drawback of the AIS is that the data structures required for maintaining frequent and candidate itemsets were not specified.

The desire to use SQL to generate frequent itemsets results in the introduction of another new algorithm known as SETM. This algorithm represent each member of the candidate/frequent itemsets in the form <TID, itemset> where TID is the unique identifier of a transaction. The Problem in SETM algorithm is that it also makes multiple passes over the database just like AIS.

Agrawal and Srikant (1994) observed that the main problem that arises in SETM is due to the number of candidates itemsets. Since, for each candidate itemset there is a TID associated with it, it requires more space to store a large number of TIDs. Furthermore, Sarawagi *et al.* (1998) has also mentioned that SETM is inefficient.

The AIS and SETM algorithms was followed by the Apriori algorithm (Agrawal and Srikant, 1994) that was shown to perform better than AIS and SETM by an order of magnitude. The most important aspect of Apriori is to completely incorporate the subset frequency based pruning optimization that is, it does not process any itemset whose subset is known to be infrequent. It utilizes a data structure called hashtree to store the counters of candidate itemsets. The main drawback in this algorithm is that it performs n passes over the database, where n is the length of the longest frequent itemset. In the Kth pass, the counts of candidate itemsets of length K (called K-itemsets) are obtained. Another drawback is that Apriori follows a tuple-by-tuple approach that is, it updates counters of candidate itemsets after reading in each transaction from the database. It hence suffers from the drawback that much redundant work (traversal of the data structure holding the counters of itemsets) is performed after each and every transaction.

Since, there are usually a large number of distinct single items in a typical transaction database and their combinations may form a very huge number of itemsets, it is challenging to develop scalable methods for mining frequent itemsets in a large transaction database. Agrawal and Srikant (1994) observed an interesting downward closure property, called Apriori, among frequent k itemsets: A k-itemset is frequent only if all of its sub-itemsets are frequent. This implies that frequent itemsets can be mined by first scanning the database to find the frequent 1-itemsets, then using the frequent 1-itemsets to generate candidate frequent 2-itemsets and check against the database to obtain the frequent 2-itemsets. This process iterates until no more frequent k-itemsets can be generated for some k. This is the essence of the Apriori algorithm (Agrawal and Srikant, 1994) and its alternative (Mannila *et al*., 1994).

Since, the Apriori algorithm was proposed, there have been extensive studies on the improvements or extensions of Apriori, e.g., partitioning technique (Savasere *et al.*, 1995), sampling approach (Toivonen, 1996) dynamic itemset counting (Brin *et al.*, 1997a). Novel method for counting the occurrences of itemsets (Tiwari *et al*., 2009), CARMA (Hidber, 1999) hashing technique (Park *et al.*, 1995a), incremental mining (Cheung *et al*., 1996), parallel and distributed mining (Park *et al.*, 1995a; Cheung *et al.*, 1996; Zaki *et al.*, 1997) and integrating mining with relational database systems (Sarawagi *et al.*, 1998). Geerts *et al.* (2001) derived a tight upper bound of the number of candidate patterns that can be generated in the level-wise mining approach. This result is effective at reducing the number of database scans. Following section describes some important variations/extensions of Apriori algorithm.

Variants of apriori

- **Partitioning technique:** The partitioning technique was introduced by Savasere *et al.* (1995), wherein the database is logically divided in to a number of disjoint partitions. This technique requires at most two passes over the database and is based on the observation that an itemset can be globally frequent over the entire database iff it is locally frequent in at least one partition. The counting strategy in this algorithm computes for each candidate itemset, a list of tids of transactions that contain the itemset. These list (also referred to as tid-lists) are computed separately for each partition and are used for efficient counting
- **Sampling approach:** This approach, proposed by Toivonen (1996) first mines a random sample of the database to obtain itemsets that are frequent within the sample. These itemsets could be considered as a representative of the actual frequent itemsets in applications where approximate mining results are sufficient. In order to obtain accurate mining results, this approach requires one or two scans over the entire database. The sampling algorithm too follows a tuple-by-tuple approach and hence, like Apriori, suffers from the above mentioned drawback
- **DIC:** The DIC algorithm (Brin *et al.*, 1997b) also known as non-level-wise algorithm. In DIC, candidates are generated and removed after every M transaction, where M is a parameter to the algorithm. Although, it is a multi-pass algorithm, it was shown to complete within two passes typically. It however,

suffers from the drawbacks of tuple-by-tuple approaches. This algorithm is considered as a closer to the sampling approach proposed by Toivonen (1996)

- **CARMA:** CARMA (Continuous Association Rule Mining Algorithm) (Hidber, 1999) brings the computation of frequent itemsets online. Being online, CARMA shows the current association rules to the user and allow the users to change the parameters, minimum support and minimum confidence, at any transaction during the first scan of the database. This is a 2-pass algorithm offering features for dynamically generating and removing candidates after each tuple of the database is processed. It was shown by Hidber (1999) that while CARMA did not perform consistently better than Apriori, its memory utilization was less by an order of magnitude

Mining frequent itemsets without candidate generation

FP-tree based algorithm: In many cases, the Apriori algorithm significantly reduces the size of candidate sets using the Apriori property. However, it can suffer from two-nontrivial costs: (1) generating a huge number of candidate sets and (2) repeatedly scanning the database and checking the candidates by pattern matching. Han *et al*. (2004) devised an FP-growth method that mines the complete set of frequent itemsets without candidate generation. FP-growth is based on the divide and-conquer principle. The first scan of the database derives a list of frequent items in which items are ordered by frequency descending order. According to the frequency-descending list, the database is compressed into a frequent pattern tree, or FP-tree, which retains the itemset association information. The FP-tree is mined by starting from each frequent length-1 pattern (as an initial suffix pattern), constructing its conditional pattern base (a sub-database, which consists of the set of prefix paths in the FP-tree co-occurring with the suffix pattern), then constructing its conditional FP-tree and performing mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree. The main problem in FP-tree is that the construction of the frequent pattern tree is a time consuming activity. Further FP-tree based approaches do not offer flexibility and reusability of computation during mining process.

The FP-growth algorithm transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix. It uses the least frequent items as a suffix, offering good selectivity. Performance studies demonstrate that the method substantially reduces search time.

There are many alternatives and extensions to the FP-growth approach, including depth-first generation of frequent itemsets by Agarwal *et al.* (2001) H-Mine, by Pei *et al.* (2001a) which explores a hyper-structure mining of frequent patterns; building alternative trees; exploring top-down and bottom-up traversal of such trees in pattern-growth mining by Liu *et al.* (2002, 2003) and an array-based implementation of prefix-tree-structure for efficient pattern growth mining by Grahne and Zhu (2003).

Mining frequent itemsets using vertical data layout: Most of the algorithms discussed earlier generate frequent itemsets from a set of transactions in horizontal data format (i.e., {TID: itemset}), where TID is a transaction- id and itemset is the set of items contained in transaction TID. Alternatively, mining can also be performed with data presented in vertical data format (i.e., {item: TID_set}). Brief introductions of algorithms that supports vertical data format are given below:

- **MaxClique:** While the above mentioned algorithms were primarily horizontal (tuple) based approaches, the MaxClique (Zaki *et al*., 1997) algorithm is designed to efficiently mine databases that are available in a vertical layout
- **Eclat:** Zaki (2000) proposed Equivalence CLASS Transformation (Eclat) algorithm by exploring the vertical data format. The first scan of the database builds the TID_set of each single item. Starting with a single item (k = 1), the frequent (k+1)-itemsets grown from a previous k-itemset can be generated according to the Apriori property, with a depth-first computation order similar to FP-growth (Han *et al*., 2004). The computation is done by intersection of the TID_sets of the frequent k-itemsets to compute the TID_sets of the corresponding (k+1)-itemsets. This process repeats, until no frequent itemsets or no candidate itemsets can be found. Besides taking advantage of the Apriori property in the generation of candidate (k + 1)-itemset from frequent k-itemsets, another merit of this method is that there is no need to scan the database to find the support of (k + 1)-itemsets (for k = 1). This is because the TID_set of each k-itemset carries the complete information required for counting such support
- **VIPER:** Unlike earlier vertical mining algorithms which were subject to various restrictions on the underlying database size, shape, contents or the mining process, the viper (Shenoy *et al*., 2000) algorithm does not have any such restrictions. It include many optimizations to enable efficient processing and was shown to outperform earlier vertical mining algorithms

MAXIMAL AND CLOSED FREQUENT PATTERN MINING ALGORITHMS

A major challenge in mining frequent patterns from a large data set is the fact that such mining often generates a huge number of patterns satisfying the min_sup threshold, especially when min_sup is set low. This is because if a pattern is frequent, each of its subpatterns is frequent as well. A large pattern will contain an exponential number of smaller, frequent sub-patterns. To overcome this problem, closed frequent pattern mining and maximal frequent pattern mining were proposed.

Definition 1: A pattern α is a closed frequent pattern in a data set D if α is frequent in D and there exists no proper super-pattern β such that β has the same support as α in D.

Definition 2: A pattern α is a maximal frequent pattern (or max-pattern) in set D if α is frequent and there exists no super-pattern β such that $\alpha \subseteq \beta$ and β is frequent in D.

For the same min_sup threshold, the set of closed frequent patterns contains the complete information regarding to its corresponding frequent patterns; whereas the set of max-patterns, though more compact, usually does not contain the complete support information regarding to its corresponding frequent patterns.

The mining of frequent closed itemsets was proposed by Pasquier *et al.* (1999), where an Apriori-based algorithm called A-Close for such mining was presented.

Other closed pattern mining algorithms include CLOSET (Pei *et al.*, 2000), CHARM (Zaki and Hsiao, 2002), CLOSET+ (Wang *et al.*, 2003a), FPClose (Grahne and Zhu, 2003) and AFOPT (Liu *et al.*, 2003).

The main challenge in closed (maximal) frequent pattern mining is to check whether a pattern is closed (maximal). There are two strategies to approach this issue: (1) to keep track of the TID list of a pattern and index the pattern by hashing its TID values. This method is used by CHARM which maintains a compact TID list called a diffset and (2) to maintain the discovered patterns in a pattern-tree similar to FP-tree. This method is exploited by CLOSET+, AFOPT and FPClose. A Frequent Itemset Mining Implementation (FIMI) workshop dedicated to the implementation methods of frequent itemset mining was reported by Goethals and Zaki (2003). Mining closed itemsets provides an interesting and important alternative to mining frequent itemsets since, it inherits the same analytical power but generates a much smaller set of results. Better scalability and interpretability is achieved with closed itemset mining.

Mining max-patterns was first studied by Bayardo (1998), where MaxMiner (Bayardo, 1998), an Apriori-based, level-wise, breadth-first search method was proposed to find max-itemset by performing superset frequency pruning and subset infrequency pruning for search space reduction.

Another efficient method MAFIA, proposed by Burdick *et al.* (2001) uses vertical bitmaps to compress the transaction id list, thus improving the counting efficiency. Yang (2004) provided theoretical analysis of the (worst-case) complexity of mining max-patterns.

PARALLEL ALGORITHMS

Patel *et al.* (2005) have proposed parallel algorithm for the mining of frequent itemsets. This is an efficient algorithm for mining frequent itemsets from those databases, whose size is very large and have high data skewness.

Other algorithms which adopt the data parallelism include CD (PDM (Park *et al.*, 1995b), DMA (Cheung *et al.*, 1996), CCPD (Zaki *et al.*, 1996) and Lattice based algorithm (Sharma *et al.*, 2007). These algorithms differ in whether further candidate pruning or efficient candidate counting techniques are employed or not.

INTERESTING PATTERNS

Several scalable methods have been developed for mining frequent patterns and closed (maximal) patterns, such mining methods often generates a huge number of frequent patterns. Sometimes it becomes essential for a user to consider only useful patterns or required interesting ones.

Many recent studies have contributed to mining interesting patterns or rules, including constraint-based mining, mining incomplete or compressed patterns and interestingness measure and correlation analysis.

Constraint-based mining: Although, a data mining process may uncover thousands of patterns from a given set of data, a particular user is interested in only a small subset of them, satisfying some user-specified constraints. Efficient mining only the patterns that satisfy user-specified constraints is called constraint-based mining. Studies have found that constraints can be categorized into various categories according to their interaction with the mining process. For example, succinct constraints can be pushed into the initial data selection process at the start of mining, anti-monotonic can be pushed deep to restrain pattern growth during mining and monotonic constraints can be checked and once satisfied, not to do more constraint checking at their further pattern growth (Ng *et al*., 1998; Lakshmanan *et al*., 1999) the push of monotonic constraints for mining correlated frequent itemsets was studied in the context of Grahne *et al*. (2000). The push of convertible constraints, such as avg() = v, can be performed by sorting items in each transaction in their value ascending or descending order for constrained pattern growth (Pei *et al*., 2001a). Since many commonly used constraints belong to one of the above categories, they can be pushed deeply into the mining process. A dual mining approach was proposed by Bucila *et al*. (2003). An algorithm, ExAnte, was proposed by Bonchi *et al*. (2003) to further prune the data search space with the imposed monotone constraints. Gade *et al*. (2004) proposed a block constraint which determines the significance of an itemset by considering the dense block formed by the pattern's items and transactions. An efficient algorithm is developed to mine the closed itemsets that satisfy the block constraints. Bonchi and Lucchese (2004) proposed an algorithm for mining closed constrained patterns by pushing deep monotonic constraints as well. Yun and Leggett (2005) proposed a weighted frequent itemset mining algorithm with the aim of pushing the weight constraint into the mining while maintaining the downward closure property.

Compressed or approximate pattern mining: To reduce the huge set of frequent patterns generated in data mining while maintain the high quality of patterns, recent studies have been focusing on mining a compressed or approximate set of frequent patterns. In general, pattern compression can be divided into two categories:

- Lossless compression
- Lossy compression

In terms of the information that the result set contains, compared with the whole set of frequent patterns.

Mining closed patterns, described as earlier is a lossless compression of frequent patterns. Mining all non-derivable frequent sets proposed by Calders and Goethals (2005) belongs to this category as well since the set of result patterns and their support information generated from these methods can be used to derive the whole set of frequent patterns. A depth-first algorithm, based on Eclat, was proposed by Calders and Goethals (2005) for mining the non-derivable itemsets.

Liu *et al*. (2006a) proposed to use a positive border with frequent generators to form a lossless representation. Lossy compression is adopted in most other compressed patterns, such as maximal patterns by Bayardo (1998), top-k most frequent closed patterns by Wang *et al*. (2003a, b), condensed pattern bases by Pei *et al*. (2002a), k-summarized patterns or pattern profiles by Afrati *et al*. (2004) and Yan *et al*. (2005a) and clustering-based compression by Xin *et al*. (2005).

For mining top-k most frequent closed patterns, a TFP algorithm is proposed to discover top-k closed frequent patterns of length no less than min_ l. TFP gradually raises the support threshold during the mining and prunes the FP-tree both during and after the tree construction phase. Due to the uneven frequency distribution among itemsets, the top-k most frequent patterns usually do not represent the most representative k patterns. Another branch of the compression work takes a "summarization" approach where the aim is to derive k representatives which cover the whole set of (closed) frequent itemsets. The k representatives provide compact compression over the collection of frequent patterns, making it easier to interpret and use. Afrati *et al*. (2004) proposed using k itemsets to approximate a collection of frequent itemsets. The measure of approximating a collection of frequent itemsets with k itemsets is defined to be the size of the collection covered by the k itemsets. Yan *et al*. (2005b) proposed a profile-based approach to summarize a set of (closed) frequent itemsets into k representatives. A profile over a set of similar itemsets is defined as a union of these itemsets, as well as item probability distribution in the supporting transactions. The highlight of profile-based approach is its ability in restoration of individual itemsets and their supports with small error. Clustering-based compression views frequent patterns as a set of patterns grouped together based on their pattern similarity and frequency support. The condensed pattern-base approach (Pei *et al*., 2002b) partitions patterns based on their support and then find the most representative pattern in each group. The representative pattern approach by Xin *et al*. (2005) clusters the set of frequent itemsets based on both pattern similarity and frequency with a tightness measure δ (called δ-cluster).

Since, real data is typically subject to noise and measurement error, it is demonstrated through theoretical results that, in the presence of even low levels of noise, large frequent itemsets are broken into fragments of logarithmic size; thus the itemsets cannot be recovered by a routine application of frequent itemset mining. Yang *et al*. (2001) proposed two error-tolerant models, termed weak Error-Tolerant Itemsets (ETI) and strong ETI. The support envelope proposed by Steinbach *et al*. (2004) is a tool for exploration and visualization of the high-level structures of association patterns. Asymmetric ETI model is proposed such that the same fraction of errors are allowed in both rows and columns. Seppänen and Mannila (2004) proposed to mine the dense itemsets in the presence of noise where the dense itemsets are the itemsets with a sufficiently large submatrix that exceeds a given density threshold of attributes present. Liu *et al.* (2006b) developed a general model for mining Approximate Frequent Itemsets (AFI) which controls errors of two directions in matrices formed by transactions and items.

Rule interestingness: Frequent itemset mining leads to the discovery of associations and correlations among items in large transaction data sets. The discovery of interesting association or correlation relationships can help in many business decision-making processes, such as catalog design, cross-marketing and customer shopping behavior analysis.

The concept of association rule was introduced together with that of frequent pattern (Agrawal *et al.*, 1993).

Let $I = \{i1, i2, . . . , im\}$ be a set of items. An association rule takes the form of $\alpha \rightarrow \beta$, where $\alpha \subset I$, $\beta \subset I$ and $\alpha \cap \beta = \varphi$ and support and confidence are two measures of rule interestingness. An association rule is considered interesting if it satisfies both a min_ sup threshold and a min_ conf threshold.

Based on the definition of association rule, most studies take frequent pattern mining as the first and the essential step in association rule mining. However, not all the association rules so generated are interesting, especially when mining at a low support threshold or mining for long patterns. To mine interesting rules, a correlation measure has been used to augment the support-confidence framework of association rules. This leads to the correlation rules of the form $\alpha > \beta$ (support, confidence, correlation). There are various correlation measures including χ^2, cosine and all_ confidence.

The problem of rule interestingness has been studied by many researchers. Piatetski-Shapiro proposed the statistical independence of rules as an interestingness measure (Piatetsky-Shapiro, 1991). Brin *et al.* (1997b) proposed lift and χ^2 as correlation measures and developed an efficient mining method. Aggarwal and Yu (1998) studied the weakness of the support-confidence framework and proposed the strongly collective itemset model for association rule generation.

Other alternatives to the support-confidence framework for assessing the interestingness of association rules are proposed by Brin *et al.* (1997b) and Ahmed *et al.* (2000).

Silverstein *et al.* (1998) studied the problem of mining causal structures over transaction databases. Some comparative studies of different interestingness measures were done by Hilderman and Hamilton (2001) and Tan *et al.* (2002). Since the probability of an item appearing in a particular transaction is usually very low, it is desirable that a correlation measure should not be influenced by null-transactions, i.e., the transactions that do not contain any of the items in the rule being examined. Tan *et al.* (2002), Omiecinski (2003) and Lee *et al.* (2003) found that all_confidence, coherence and cosine are null-invariant and are thus good measures for mining correlation rules in transaction databases.

Shekar and Natarajan (2004) proposed a data-driven approach for assessing the interestingness of association rules, which is evaluated by using relatedness based on relationships between item pairs.

Blanchard *et al.* (2005) designed a rule interestingness measure, Directed Information Ratio, based on information theory. This measure could filter out the rules whose antecedent and consequent are negatively correlated and the rules which have more counter examples than examples.

Gionis *et al.* (2006) recently proposed a new significance assessment that not only depends on the specific attributes, but also on the dataset as a whole, which is often missed by many existing methods such as χ^2 tests.

Studies were also conducted on mining interesting or unexpected patterns compared with user's prior knowledge. Wang *et al.* (2003b) defined a preference model which captures the notion of unexpectedness. An algorithm was proposed for mining all unexpected rules which satisfy user-specified minimum unexpectedness significance and unexpectedness strength. In Jaroszewicz and Scheffer (2005) and Jaroszewicz and Simovici (2004) user's prior knowledge is expressed by a Bayesian network. The interestingness of an itemset is defined as the absolute difference between its support estimated from the data and from the Bayesian network. User's feedback on interestingness could also guide the discovery of interesting patterns (Xin *et al.*, 2006).

Frequent pattern mining in high-dimensional databases: The growth of bioinformatics has resulted in datasets with new characteristics. Microarray and mass spectrometry technologies, which are used for measuring gene expression level and cancer research respectively, typically generate only tens or hundreds of very high-dimensional data (e.g., in 10,000-100,000 columns). If we take each sample as a row (or TID) and each gene as a column (or item), the table becomes extremely wide in comparison with a typical business transaction table. Such datasets pose a great challenge for existing (closed) frequent itemset mining algorithms, since, they have an exponential number of combinations of items with respect to the row length. Pan *et al.* (2003) proposed CARPENTER, a method for finding closed patterns in high-dimensional biological datasets, which integrates the advantages of vertical data formats and pattern growth methods. By converting data into vertical data format {item: TID_set}, the TID_set can be viewed as rowset and the FP-tree so constructed can be viewed as a row enumeration tree. CARPENTER conducts a depth-first traversal of the row enumeration tree and checks each rowset corresponding to the node visited to see whether it is frequent and closed.

Pan *et al.* (2004) proposed COBBLER, to find frequent closed itemset by integrating row enumeration with column enumeration. Its efficiency has been demonstrated in experiments on a data set with high dimension and a relatively large number of rows.

Liu *et al.* (2006c) proposed TD-Close to find the complete set of frequent closed patterns in high dimensional data. It exploits a new search strategy, top-down mining, by starting from the maximal rowset, integrated with a novel row enumeration tree, which makes full use of the pruning power of the min_sup threshold to cut down the search space. Furthermore, an effective closeness-checking method is also developed that avoids scanning the dataset multiple times.

Even with various kinds of enhancements, the above frequent, closed and maximal pattern mining algorithms still encounter challenges at mining rather large (called colossal) patterns, since the process will need to generate an explosive number of smaller frequent patterns. Colossal patterns are critical to many applications, especially in domains like bioinformatics. Zhu *et al.* (2007) investigated a novel mining approach, called Pattern-Fusion, to efficiently find a good approximation to colossal patterns. With Pattern-Fusion, a colossal pattern is discovered by fusing its small fragments in one step, whereas the incremental pattern-growth mining strategies, such as those adopted in Apriori and FP-growth, have to examine a large number of mid-sized ones. This property distinguishes Pattern-Fusion from existing frequent pattern mining approaches and draws a new mining methodology. Further extensions on this methodology are currently under investigation.

INCREMENTAL ALGORITHMS

Here, we provide an overview of the algorithms that have been developed over the last few years for incremental association rule mining.

The FUP algorithm: The FUP (Fast Update algorithm) (Cheung *et al*., 1996) represents the first work in this field of incremental mining. It operates on an iterative basis and in each iteration makes a complete scan of the current database. In each scan, the increment is processed first and the results obtained are used to guide the mining of the original database DB.

An important point to note about the FUP algorithm is that it requires k passes over the entire databse, where k is the cardinality of the longest frequent itemset. Further, it does not generate the mining results for solely the increment.

In the first pass over the increment, all the 1-itemsets are considered as candidates. At the end of this pass, the complete supports of the candidates that happen to be also frequent in DB are known. Those which have the minimum support are retained in $L^{DB \cup db}$. Among the other candidates only those which were frequent in db can become frequent due to the following theorem (Theorem 1).

Theorem 1: An itemset can be present in $F_{DB \cup db}$ only if it is present in either F_{DB} or F_{db} (or both). Hence, they are identified and the previous database DB is scanned to obtain their overall supports, thus obtaining the set of all frequent 1-itemsets. The candidates for the next pass are calculated using the AprioriGen function and the process repeats in this manner until all the frequent itemsets have been identified.

After FUP, algorithms that utilized the negative border information were proposed independently by Thomas *et al.* (1997) with the goal of achieving more efficiency in the incremental mining process. In this approach, itemsets that were originally in the negative border of the frequent itemsets and later become frequent after the database has been updated are referred to as promoted borders.

The borders algorithm: The original borders algorithm computes the entire negative border closure at one shot and then makes a scan of the entire database to compute the counts of itemsets in the closure. This could potentially result in a candidate explosion problem.

A new version of the Borders algorithm was proposed by Aumann and Lindell (1999). This version goes to the other extreme of the closure computation and makes one scan of the entire database for each layer of the negative border closure. As observed by many researchers this strategy could result in a significant increase in the number of database passes and may therefore be problematic for large databases.

Another new algorithm was proposed to handle multi-support mining. The applicability of this algorithm, however, is limited to the very special case of zero-size increments, that is, where the database has not changed at all between the previous and the current mining.

Finally, like FUP, Borders also does not generate the mining results for solely the increment.

The TBAR algorithm: The TBAR algorithm (Thomas *et al*., 1997) initially completely mines the increment db by applying the Apriori algorithm. We expect this strategy to be inefficient for large increments since the previous mining results are not used at all in this mining process.

Next, it adopts an approach similar to borders in that it computes the entire negative border closure. However, since the results of mining the increment are available at this time, this information could be used to prune more candidates from the closure- after computing each level of the closure, itemsets that are infrequent in the increment are excluded from further candidate generation. Therefore, unlike Borders, the candidate explosion problem is unlikely to occur. However, even with this pruning, there are likely to be too many unnecessary candidates in TBAR, especially for skewed increments since it relies solely on the increment for pruning.

Other algorithms: It was briefly mentioned by (Hidber, 1999) that CARMA, a first- time mining algorithm could be also applied for incremental mining. Although, the algorithm is a novel and efficient approach for first-time mining, it is noted that it suffers from the following drawbacks when applied to incremental mining: It does not maintain negative border information and hence will need to access the original database DB if there are any locally frequent itemsets in the increment, even though these itemsets may not be globally frequent.

The shrinking support intervals which CARMA maintains for candidate itemsets are not likely to be tight for itemsets that become potentially frequent while processing the increment.

An incremental mining algorithm, called MLUp, for updating multi-level association rules over a taxonomy hierarchy was presented by Cheung *et al.* (1996). It uses a different minimum support threshold for each level of the hierarchy and restricts its attention to deriving intra-level rules, that is, rules within each level.

EXTENSIONS OF ASSOCIATION RULE MINING

Multilevel, multidimensional and quantitative rule generation: Since, data items and transactions are (conceptually) organized in multilevel and/or multidimensional space, it is natural to extend mining frequent itemsets and their corresponding association rules to multi-level and multidimensional space. Multilevel association rules involve concepts at different levels of abstraction, whereas multidimensional association rules involve more than one dimension or predicate.

In many applications, it is difficult to find strong associations among data items at low or primitive levels of abstraction due to the sparsity of data at those levels. On the other hand, strong associations discovered at high levels of abstraction may represent common sense knowledge. Therefore, multilevel association rules provide sufficient flexibility for mining and traversal at multiple levels of abstraction. Multilevel association rules can be mined efficiently using concept hierarchies under a support-confidence framework. For example, if the min_sup threshold is uniform across multi-levels, one can first mine higher-level frequent itemsets and then mine only those itemsets whose corresponding high-level itemsets are frequent (Han and Fu, 1995). Moreover, redundant rules can be filtered out if the lower-level rules can essentially be derived based on higher-level rules and the corresponding item distributions. Efficient mining can also be derived if min_sup varies at different levels. Such methodology can be extended to mining multidimensional association rules when the data or transactions are located in multidimensional space, such as in a relational database or data warehouse (Kamber *et al*., 1997).

Our previously discussed frequent patterns and association rules are on discrete items, such as item name, product category and location. However, one may like to find frequent patterns and associations for numerical attributes, such as salary, age and scores. For numerical attributes, quantitative association rules can be mined, with a few alternative methods, including exploring the notion of partial completeness, by Srikant and Agrawal (1996).

Mining quantitative association rules based on a statistical theory to present only those that deviate substantially from normal data was studied by Aumann and Lindell (1999). Zhang *et al.* (2004a, b) considered mining statistical quantitative rules. Statistical quantitative rules are quantitative rules in which the right hand side of a rule can be any statistic that is computed for the segment satisfying the left hand side of the rule.

SEQUENTIAL MINING

A sequence database consists of ordered elements or events, recorded with or without a concrete notion of time. There are many applications involving sequence data, such as customer shopping sequences, Web click streams and biological sequences. Sequential pattern mining, the mining of frequently occurring ordered events or subsequences as patterns, was first introduced by Agrawal and Srikant (1995) and has become an important problem in data mining.

Generalized Sequential Patterns (GSP), a representative Apriori-based sequential pattern mining algorithm, proposed by Srikant and Agrawal (1996), uses the downward-closure property of sequential patterns and adopts a multiple pass, candidate generate-and-test approach. GSP also generalized their earlier notion in Agrawal and Srikant (1995) to include time constraints, a sliding time window and user-defined taxonomies.

Zaki (2001) developed a vertical format-based sequential pattern mining method called SPADE, which is an extension of vertical format-based frequent itemset mining methods, like Eclat and CHARM (Zaki, 1998; Zaki and Hsiao, 2002). In vertical data format, the database becomes a set of tuples of the form <itemset: (sequence_ID, event_ID)>. The set of ID pairs for a given itemset forms the ID_list of the itemset. To discover the length-k sequence, SPADE joins the ID_lists of any two of its length-(k-1) subsequences. The length of the resulting ID_list is equal to the support of the length-k sequence. The procedure stops when no frequent sequences can be found or no sequences can be formed by such joins. The use of vertical data format reduces scans of the sequence database. The ID_lists carry the information necessary to compute the support of candidates. However, the basic search methodology of SPADE and GSP is breadth-first search and Apriori pruning. Both algorithms have to generate large sets of candidates in order to grow longer sequences.

PrefixSpan, a pattern-growth approach to sequential pattern mining, was developed by Pei *et al.* (2001b, 2004). PrefixSpan works in a divide-and-conquer way. The first scan of the database derives the set of length-1 sequential patterns. Each sequential pattern is treated as a prefix and the complete set of sequential patterns can be partitioned into different subsets according to different prefixes. To mine the subsets of sequential patterns, corresponding projected databases are constructed and mined recursively.

A performance comparison of GSP, SPADE and PrefixSpan shows that PrefixSpan has the best overall performance (Pei *et al*., 2004). SPADE, although weaker than PrefixSpan in most cases, outperforms GSP. The comparison also found that when there is a large number of frequent subsequences, all three algorithms run slowly. The problem can be partially solved by closed sequential pattern mining, where closed subsequences are those sequential patterns containing no super sequence with the same support.

MISCELLANEOUS WORK IN DIFFERENT DOMAINS

Frequent patterns, reflecting strong associations among multiple items or objects, capture the underlying semantics in data. They were successfully applied to inter-disciplinary domains beyond data mining. A brief introduction of some important domains is presented below.

Discovery of spatiotemporal knowledge and patterns: A spatial database stores a large amount of space-related data, such as maps, preprocessed remote sensing or medical imaging data and VLSI chip layout data. A spatiotemporal database stores time-related spatial data, such as weather dynamics, moving objects, or regional developments. Spatial data mining refers to the extraction of knowledge, spatial relationships, or other interesting patterns from spatial data. Similarly, spatiotemporal data mining is to find spatiotemporal knowledge and patterns.

Due to the complexity of spatiotemporal data objects and their relationships as well as their associated high computational cost, it is costly to mine spatiotemporal frequent patterns in spatiotemporal data. One important methodology that may substantially reduce the computational cost is progressive refinement (Koperski and Han, 1995), which performs rough computation at a coarse resolution and refines the results only for those promising candidates at finer resolutions. Koperski and Han (1995) proposed such a methodology at mining spatial association rules (or frequent patterns).

Li *et al.* (2006) show that even for mining outliers in massive moving object data sets, one can find movement fragment patterns by spatial overlay and such movement fragments can be taken as motifs for further identification of outliers by motif-based classification.

Discovery of knowledge and patterns from multimedia data: A multimedia database system stores and manages a large collection of multimedia data, such as audio, video, image, graphics, speech, text, document and hypertext data. Multimedia data mining refers to the discovery of patterns and knowledge from multimedia data.

Frequent pattern analysis in multimedia data plays a similar important role in multimedia data mining. To mine frequent patterns in multimedia data, each image object can be treated as a transaction and frequently occurring patterns among different images can be discovered. Notice that an image may contain multiple objects, each with many features such as color, shape, texture, keyword and spatial location, so there could be many possible associations. Moreover, since a picture containing multiple recurrent objects is an important feature in image analysis, recurrence of the same object should be considered as important in frequent pattern analysis. Furthermore, spatial relationships among different objects in an image are also considered crucial in image analysis. Thus all these factors will be considered in multimedia frequent pattern mining. Zaiane *et al*. (2000) takes those factors into consideration and developed a progressive refinement algorithm for mining multimedia associations.

Discovery of knowledge and patterns from stream data: Unlike traditional data sets, stream data flow in and out of a computer system continuously and with varying update rates. It may be impossible to store an entire data stream or to scan through it multiple times due to its tremendous volume. To discover knowledge or patterns from data streams, it is necessary to develop single-scan and on-line mining methods.

For mining frequent items and itemsets on stream data, Manku and Motwani proposed sticky sampling and lossy counting algorithms for approximate frequency counts over data streams (Manku and Motwani, 2002).

Karp *et al.* (2003) proposed a counting algorithm for finding frequent elements in data streams.

Chang and Lee (2003) proposed an algorithm for finding recent frequent itemsets adaptively over an online data stream by decaying the effect of old transactions.

Yu *et al*. (2004) proposed an FDPM algorithm for mining frequent itemsets over data streams with a false-negative oriented approach.

It is argued by Yu *et al*. (2004) that, compared with the false-positive mining approach (e.g., lossy counting), the false-negative approach can effectively mine frequent itemsets with a bound of memory consumption, while in the false-positive approach, the number of false-positive frequent itemsets could increase exponentially, which makes the mining intractable.

Lin *et al*. (2005) proposed an algorithm for mining frequent itemsets from data streams based on a time-sensitive sliding window.

Discovery of knowledge and patterns from web data: There are three different types of web mining: web content mining, web structure mining and web usage mining. Web content mining is a knowledge discovery task of finding information within web pages, while web structure mining aims to discover knowledge hidden in the structures linking web pages. Web usage mining is focused on the analysis of users' activities when they browse and navigate through the Web. Classical examples of web usage mining include, but not limited to, user grouping (users that often visit the same set of pages), page association (pages that are visited together) and sequential click through analysis (the same browse and navigation orders that are followed by many users).

Association rules discovered for pages that are often visited together can reveal user groups (Eirinaki and Vazirgiannis 2003) and cluster web pages. Web access patterns via association rule mining in web logs were proposed by Chen *et al.* (1996), Pei *et al.* (2000), Srivastava *et al.* (2000) and Punin *et al.* (2001). Sequential pattern mining in web logs could find browse and navigation orders (i.e., pages that are accessed immediately after another), which might be used to refine cache design and web site design. More complicated patterns such as frequent tree-like traversal patterns were examined by Chen *et al.* (1996) and Nanopoulos and Manolopoulos (2001).

Discovery of knowledge and patterns from software bugs: Performing analysis on the executions of a buggy software program is essentially a data mining process. It is interesting to observe that frequent pattern mining has started playing an important role in software bug detection and analysis.

Many interesting methods have been developed to trace crashing bugs, such as memory violation and core dumps in various aspects. However, it is still difficult to analyze non-crashing bugs such as logical errors.

PR-Miner (Li and Zhou, 2005) uses frequent pattern mining to extract application-specific programming rules from source code. A violation of these rules might indicate a potential software bug.

Frequent pattern mining has also been successfully applied for mining block correlations in storage systems. Based on CloSpan (Yan *et al*., 2003) an algorithm called C-Miner, by Li *et al.* (2004a) was proposed to mine frequent sequential patterns of correlated blocks from block access traces.

Recent developments: In 2006, Song and Rajasekaran (2006) have proposed an algorithm for mining complete frequent itemsets known as TM (Transaction mapping) algorithm (Song and Rajasekaran, 2006). In this algorithm, transaction ids of each itemset are mapped and compressed to continuous transaction intervals in a different space and the counting of itemsets is performed by intersecting these interval lists in a depth-first order along the lexicographic tree. They have shown that TM outperforms two popular algorithms namely FP-growth and Eclat.

Further, Lucchese *et al.* (2006) proposes a new scalable algorithm for discovering closed frequent itemsets, alossless and condensed representation of all the frequent itemsets that can be mined from a transactional database. This algorithm (Lucchese *et al.*, 2006) exploits a divide-and-conquer approach and a bitwise vertical representation of the database and adopts a particular visit and partitioning strategy of the search space. Algorithm utilizes a new effective and memory efficient pruning technique, which, unlike other previous proposals, does not require the whole set of closed patterns mined so far to be kept in main memory.

A novel fast mining algorithm was proposed by (Sun and Bai, 2008) for mining association rules. In this algorithm (Sun and Bai, 2008) introduces a new measure called W-Support and focus of algorithm is only on the high quality rules.

CHALLENGING ISSUES

Discussion under previous sections reveals that a lot of attention was focus on the performance and scalability of the algorithms, but not enough attention was given to the issues related to ease, flexibility and reusability for generating association rules. Literature also reveals that more studies have been carried out on frequent pattern generation. As mentioned earlier, identifying frequent itemsets is computationally expensive process. Counting the occurrences of itemsets requires a considerable amount of processing time. As a consequence, numbers of efficient algorithms are proposed in literature for mining the frequent itemsets. It is noticed that, most of the algorithms for discovering frequent patterns available in the literature require multiple passes over the database resulting in a large number of disk reads and placing a huge burden on the I/O subsystem. This calls for the introduction of mining algorithms that offers single database scan.

Although, various algorithms are available that can help reveal patterns and relationships, it does not tell the user the value or significance of these patterns.

Most of the algorithms available in the literature do not offer flexibility for testing the validity of meta rules.

Algorithms are available for maintaining the association rules due to addition or deletion of transactions in the database. However, algorithms are not available for mining incremental rules due to addition of more items.

There is a requirement for the development of parallel and/or distributed algorithms in order to speed up the computation activity as identification of frequent itemsets is computationally expensive task. This will result in overall performance improvement.

Most of the algorithms available in the literature for mining frequent itemsets do not offer flexibility for reusing the computation during mining process.

Much research is still needed to substantially reduce the size of derived patterns and enhance the quality of retained patterns (compact high quality pattern set).

Some serious limitations of current association rule mining algorithms and several related issues motivate continued studies and research in to this area. One of the strong motivating factors for the research in this area is to enhance ease, flexibility, efficiency and reusability during mining process.

CONCLUSION

Association rule mining has recently gained considerable prominence in the data mining community because of its capability of being used as an important tool for knowledge discovery and its applicability in other data mining tasks such as clustering and classification. Association rules are of interest to both database community and data mining users. We have provided a survey of previous researches done in this field and identify some important gaps available in the current knowledge.

REFERENCES

Afrati, F.N., A. Gionis and H. Mannila, 2004. Approximating a collection of frequent sets. Proceedings of the 2004 ACM SIGKDD International Conference Knowledge Discovery in Databases, Aug. 22-25, Seattle, WA., USA., pp: 12-19.

Agrawal, R., T. Imielinski and A. Swami, 1993. Mining association rules between sets of items in large databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, May 25-28, ACM, New York, USA., pp: 207-216.

Agrawal, R. and R. Srikant, 1994. Fast algorithms for mining association rules. Proceedings of the 20th International Conference on Very Large Data Bases, Sept. 12-15, San Francisco, CA., USA., pp: 487-499.

Agrawal, R. and R. Srikant, 1995. Mining sequential patterns. Proceedings of the 11th International Conference on Data Engineering, March 6-10, Taipei, Taiwan, pp: 3-14.

Aggarwal, C.C. and P.S. Yu, 1998. A new framework for itemset generation. Proceedings of the 17th ACM Symposium on Principles of Database Systems, June 1-4, Seattle, WA., pp: 18-24.

Agrawal, R., J. Gehrke, D. Gunopulos and P. Raghavan, 1998. Automatic subspace clustering of high dimensional data for data mining applications. Proceedings of ACM SIGMOD International Conference on Management of Data, June 1-4, ACM Press, New York, pp: 94-105.

Agarwal, R.C., C. Aggarwal and V.V.V. Prasad, 2001. A tree projection algorithm for generation of frequent item sets. J. Parallel Distributed Comput., 61: 350-371.

Ahmed, K.M., N.M. El-Makky and Y. Taha, 2000. A note on beyond market basket: Generalizing association rules to correlations. SIGKDD Explorat., 1: 46-48.

Aumann, Y. and Y. Lindell, 1999. A statistical theory for quantitative association rules. Proceedings of the 1999 International Conference on Knowledge Discovery and Data Mining, Aug. 15-18, San Diego, CA., USA., pp: 261-270.

Bayardo, R.J., 1998. Efficiently mining long patterns from databases. Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, June 1-4, New York, USA., pp: 85-93.

Blanchard, J., F. Guillet, R. Gras and H. Briand, 2005. Using information-theoretic measures to assess association rule interestingness. Proceedings of the 2005 International Conference on Data Mining, Nov. 27-30, Houston, TX., pp: 66-73.

Bonchi, F., F. Giannotti, A. Mazzanti and D. Pedreschi, 2003. Exante: Anticipated data reduction in constrained pattern mining. Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Sept. 22-26, Cavtat, Dubrovnik, Croatia, pp: 59-70.

Bonchi, F. and C. Lucchese, 2004. On closed constrained frequent pattern mining. Proceedings of the 2004 International Conference on Data Mining, Nov. 1-4, Brighton, UK., pp: 35-42.

Brin, S., R. Motwani and C. Silverstein, 1997a. Beyond market basket: Generalizing association rules to correlations. Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, May 11-15, Tucson, AZ., pp: 265-276.

Brin, S., R. Motwani, J.D. Ullman and S. Tsur, 1997b. Dynamic itemset counting and implication rules for market basket data. Proc. 1997 ACM SIGMOID Int. Conf. Manage. Data, 26: 255-264.

Bucila, C., J. Gehrke, D. Kifer and W. White, 2003. DualMiner: A dual-pruning algorithm for itemsets with constraints. Data Min. Knowl. Discov., 7: 241-272.

Burdick, D., M. Calimlim and J. Gehrke, 2001. MAFIA: A maximal frequent itemset algorithm for transactional databases. Proceedings of the 2001 International Conference on Data Engineering, (ICDE'01), Heidelberg, Germany, pp: 443-452.

Calders, T. and B. Goethals, 2005. Depth-first non-derivable itemset mining. Proc. SIAM Int. Conf. Data Min., 119: 250-261.

Chang, J. and W. Lee, 2003. Finding recent frequent itemsets adaptively over online data streams. Proceedings of the 2003 International Conference on Knowledge Discovery and Data Mining, Aug. 24-27, Washington, DC., USA., pp: 487-492.

Chen, M.S., J.S. Park and P.S. Yu, 1996. Data mining for path traversal patterns in a web environment. Proc. Int. Conf. Distrib. Comput. Syst., 16: 385-392.

Cheung, D.W., J. Han, V.T. Ng and C.Y. Wong, 1996. Maintenance of discovered association rules in large databases: An incremental updating technique. Proceedings of International Conference on Data Engineering, Feb. 26-Mar. 1, New Orleans, Louisiana, pp: 106-114.

Eirinaki, M. and M. Vazirgiannis, 2003. Web mining for web personalization. ACM Trans. Internet Technol., 3: 1-27.

Gade, K., J. Wang and G. Karypis, 2004. Efficient closed pattern mining in the presence of tough block constraints. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 22-25, Seattle, WA., pp: 138-147.

Geerts, F., B. Goethals and J. Bussche, 2001. A tight upper bound on the number of candidate patterns. Proceedings of the 2001 International Conference on Data Mining, Nov. 29-Dec. 2, San Jose, CA., pp: 155-162.

Gionis, A., H. Mannila, T. Mielikäinen and P. Tsaparas, 2006. Assessing data mining results via swap randomization. Proceedings of the 2006 ACM SIGKDD International Conference on Knowledge Discovery in Databases, Aug. 22-23, Philadelphia, PA., pp: 167-176.

Goethals, B. and M. Zaki, 2003. An introduction to workshop on frequent itemset mining implementations. Proceedings of the 2003 ICDM International Workshop on Frequent Itemset Mining Implementations, Dec. 19-19, Melbourne, FL., pp: 1-13.

Grahne, G., L. Lakshmanan and X. Wang, 2000. Efficient mining of constrained correlated sets. Proceedings of the 2000 International Conference on Data Engineering, Feb. 28-March 3, San Diego, CA., pp: 512-521.

Grahne, G. and J. Zhu, 2003. Efficiently using prefix-trees in mining frequent itemsets. Proceedings of the 2003 ICDM International Workshop on Frequent Itemset Mining Implementations, (IWFIMI'03), Melbourne, FL., pp: 123-132.

Han, J. and Y. Fu, 1995. Discovery of multiple-level association rules from large databases. Proceedings of the 21st international Conference on Very Large Data Bases, Sept. 11-15, Zurich, Switzerland, pp: 420-431.

Han, J., J. Pei, Y. Yin and R. Mao, 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining Knowledge Discovery, 8: 53-87.

Hidber, C., 1999. Online association rule mining. ACM SIGMOD Rec., 28: 145-156.

Hilderman, R.J. and H.J. Hamilton, 2001. Knowledge Discovery and Measures of Interest. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Jaroszewicz, S. and D.A. Simovici, 2004. Interestingness of frequent itemsets using Bayesian networks as background knowledge. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 22-25, Seattle, WA., pp: 178-186.

Jaroszewicz, S. and T. Scheffer, 2005. Fast discovery of unexpected patterns in data relative to a Bayesian network. Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 21-24, Chicago, IL., pp: 118-127.

Kamber, M., J. Han and J.Y. Chiang, 1997. Metarule-guidedmining of multi-dimensional association rules using data cubes. Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, Aug. 14-17, Newport Beach, CA., pp: 207-210.

Karp, R.M., C.H. Papadimitriou and S. Shenker, 2003. A simple algorithm for finding frequent elements in streams and bags. ACM Trans. Database Syst., 28: 51-55.

Koperski, K. and J. Han, 1995. Discovery of spatial association rules in geographic information databases. Proceedings of the 4th International Symposium on Advances in Spatial Databases, Aug. 6-9, Portland, ME., pp: 47-66.

Lakshmanan, L.V.S., R. Ng, J. Han and A. Pang, 1999. Optimization of constrained frequent set queries with 2-variable constraints. ACM SIGMOD Rec., 28: 157-168.

Lee, Y.K., W.Y. Kim, Y.D. Cai and J. Han, 2003. CoMine: Efficient mining of correlated patterns. Proceedings of the 3rd IEEE International Conference on Data Mining, Nov. 1-4, Melbourne, FL., pp: 581-584.

Li, Z., S. Lu, S. Myagmar and Y. Zhou, 2004a. CP-Miner: A tool for finding copy-paste and related bugs in operating system code. Proceedings of the USENIX Symposium on Operating Systems Design and Implementation, (SOSDI'04), San Francisco, CA., pp: 289-302.

Li, Z., Z. Chen, S.M. Srinivasan and Y. Zhou, 2004b. C-Miner: Mining block correlations in storage systems. Proceedings of the 3rd USENIX Conference on File and Storage Technologies, March 31, San Francisco, CA., pp: 173-186.

Li, Z. and Y. Zhou, 2005. PR-Miner: Automatically extracting implicit programming rules and detecting violations in large software code. Proceedings of the 10th European Software Engineering Conference and13th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Sept. 5-9, Lisbon, Portugal, pp: 306-315.

Li, X., J. Han and S. Kim, 2006. Motion-alert: Automatic anomaly detection in massive moving objects. Proceedings of the IEEE International Conference on Intelligence and Security Informatics, May 23-24, San Diego, CA., pp: 166-177.

Lin, C.H., D.Y. Chiu, Y.G. Wu and A.L.P. Chen, 2005. Mining frequent itemsets from data streams with a time-sensitive sliding window. Proc. SIAM Int. Conf. Data Min., 119: 68-79.

Liu, J., Y. Pan, K. Wang and J. Han, 2002. Mining frequent item sets by opportunistic projection. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery in Databases, July 23-26, Edmonton, Canada, pp: 239-248.

Liu, G., H. Lu, W. Lou and J.X. Yu, 2003. On computing, storing and querying frequent patterns. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 24-27, Washington, DC., pp: 607-612.

Liu, G., J. Li, L. Wong and W. Hsu, 2006a. Positive borders or negative borders: How to make lossless generator based representations concise? Proceedings of the 2006 SIAM International Conference on Data Mining, (ICDM'06), Bethesda, MD., pp: 467-471.

Liu, H., J. Han, D. Xin and Z. Shao, 2006b. Mining frequent patterns on very high dimensional data: A top down row enumeration approach. Proceedings of the 2006 SIAM International Conference on Data Mining, (ICDM'06), Bethesda, MD., pp: 280-291.

Liu, J., S. Paulsen, X. Sun, W. Wang, A. Nobel and J. Prins, 2006c. Mining approximate frequent itemsets in the presence of noise: Algorithm and analysis. Proceedings of the 2006 SIAM International Conference on Data Mining, April 20-22, Bethesda, MD., pp: 405-416.

Lucchese, C., S. Orlando and R. Perego, 2006. Fast and memory efficient mining of frequent closed Itemsets. IEEE Trans. Knowl. Data Eng., 18: 21-36.

Manku, G.S. and R. Motwani, 2002. Approximate frequency counts over data streams. Proceedings of the 28th international Conference on Very Large Databases, Aug. 20-23, ACM New York, USA., pp: 346-357.

Mannila, H., H. Toivonen and A. Inkeri Verkamo, 1994. Efficient algorithms for discovering association rules. Proceedings of the AAAI Workshop on Knowledge Discovery in Databases, (KDD-94), IEEE, pp: 181-192.

Nanopoulos, A. and Y. Manolopoulos, 2001. Mining patterns from graph traversals. Data Knowl. Eng., 37: 243-266.

Ng, R., L.V.S. Lakshmanan, J. Han and A. Pang, 1998. Exploratory mining and pruning optimizations of constrained associations rules. ACM SIGMOD Rec., 27: 13-24.

Omiecinski, E, 2003. Alternative interestmeasures formining associations. IEEE Trans. Knowl. Data Eng., 15: 57-69.

Pan, F., G. Cong, A.K.H. Tung, J. Yang and M. Zaki, 2003. CARPENTER: Finding closed patterns in long biological datasets. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 24-27, Washington, DC., pp: 637-642.

Pan, F., A.K.H. Tung, G. Cong and X. Xu, 2004. COBBLER: Combining column and row enumeration for closed pattern discovery. Proceedings of the 2004 International Conference on Scientific and Statistical Database Management, (ICSSDM'04), Santorini Island, Greece, pp: 21-30.

Park, J.S., M.S. Chen and P.S. Yu, 1995a. An effective hash-based algorithm for mining association rules. ACM SIGMOD Rec., 24: 175-186.

Park, J.S., M.S. Chen and P.S. Yu, 1995b. Efficient parallel mining for association rules. Proceedings of the 4th International Conference on Information and Knowledge Management, Nov. 29-Dec. 2, Baltimore, MD., pp: 31-36.

Pasquier, N., Y. Bastide, R. Taouil and L. Lakhal, 1999. Discovering frequent closed itemsets for association rules. Proceedings of the 7th International Conference on Database Theory, Jan. 10-12, Jerusalem, Israel, pp: 398-416.

Patel, P., S.S. Rana and K.R. Pardasani, 2005. Model for load balancing on processors in parallel mining of frequent itemsets. Am. J. Applied Sci., 2: 926-931.

Pei, J., J. Han, B. Mortazavi-Asl and H. Zhu, 2000. Mining access patterns efficiently from web logs. Proceedings of the 2000 Pacific-Asia Conference on Knowledge Discovery and Data Mining: Current Issues and New Applications, April 18-20, Kyoto, Japan, pp: 396-407.

Pei, J., J. Han and L.V.S. Lakshmanan, 2001a. Mining frequent itemsets with convertible constraints. Proceedings of the 17th International Conference on Data Engineering, April 2-6, Heidelberg, Germany, pp: 433-332.

Pei, J., J. Han, M.A. Behzad, P. Helen, Q. Chen and M.C. Hsu, 2001b. Prefix span: Mining sequential patterns efficiently by prefix-projected pattern growth. Proceedings of the 17th International Conference on Data Engineering, (ICDE'01), Heidelberg, pp: 215-224.

Pei, J., G. Dong, W. Zou and J. Han, 2002a. Oncomputing condensed frequent pattern bases. Proceedings of the 2002 International Conference on Data Mining, Dec. 9-12, Maebashi, Japan, pp: 378-385.

Pei, J., J. Han and W. Wang, 2002b. Constraint-based sequential pattern mining in large databases. Proceeding of the 2002 International Conference on Information and Knowledge Management, Nov. 4-9, McLean, VA., pp: 18-25.

Pei, J., J. Han, B. Mortazavi-Asl, J. Wang and H. Pinto *et al.*, 2004. Mining sequential patterns by pattern-growth: The prefixspan approch. IEEE Trans. Knowledge Data Eng., 16: 1424-1440.

Piatetsky-Shapiro, G., 1991. Notes of AAAI'91 Workshop Knowledge Discovery in Databases (KDD'91). AAAI/MIT Press, Anaheim, CA.

Sarawagi, S., S. Thomas and R. Agrawal, 1998. Integrating association rule mining with relational database systems: Alternatives and implications. ACM SIGMOD Rec., 27: 343-354.

Savasere, A., E. Omieccinski and S. Navathe, 1995. An efficient algorithm for mining association rules in large databases. Proceedings of the 21st International Conference on Very Large Databases, Sept. 11-15, Zurich, Switzerland, pp: 432-443.

Seppänen, J. and H. Mannila, 2004. Dense itemsets. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 22-25, Seattle, WA., pp: 683-688.

Sharma, S., A. Tiwari, S. Sharma and K.R. Pardasani, 2007. Design of algorithm for frequent pattern discovery using lattice approach. Asian J. Inform. Manage., 1: 11-18.

Shekar, B. and R. Natarajan, 2004. A transaction-based neighbourhood-driven approach to quantifying interestingness of assoication rules. Proceedings of the 4th IEEE International Conference on Data Mining, Nov. 1-4, Brighton, UK., pp: 194-201.

Shenoy, P., J. Haritsa, S. Sudarshan, G. Bhalotia, M. Bawa and D. Shah, 2000. Turbo-charging vertical mining of large databases. Proceedings of ACM SIGMOD International Conference on Management of Data, MAY 16-18, Dallas, TX., pp: 22-33.

Silverstein, C., S. Brin, R. Motwani and J.D. Ullman, 1998. Scalable techniques for mining causal structures. Proceedings of the 24rd International Conference on Very Large Data Bases, Aug. 24-27, San Francisco, CA., USA., pp: 594-605.

Song, M. and S. Rajasekaran, 2006. A transaction mapping algorithm for frequent itemsets mining. IEEE Trans. Knowl. Data Eng., 18: 472-481.

Srikant, R. and R. Agrawal, 1996. Mining sequential patterns: Generalizations and performance improvements. Proc. Int. Conf. Extend. Database Technol., 1057: 3-17.

Srivastava, J., R. Cooley, M. Deshpande and P.N. Tan, 2000. Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorat., 1: 12-23.

Steinbach, M., P. Tan and V. Kumar, 2004. Support envelopes: A technique for exploring the structure of association patterns. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery in Databases, Aug. 22-25, Seattle, WA., pp: 296-305.

Sun, K. and F. Bai, 2008. Mining weighted association rules without preassigned weighted. IEEE Trans. Knowl. Data Eng., 20: 489-495.

Tan, P.N., V. Kumar and J. Srivastava, 2002. Selecting the right interestingness measure for association patterns. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery in Databases, July 23-26, Edmonton, Canada, pp: 32-41.

Thomas, S., S. Bodagala, K. Alsabti and S. Ranka, 1997. An efficient algorithm for the incremental updation of association rules in large databases. Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining, Aug. 14-17, Newport Beach, CA., pp: 263-266.

Tiwari, A., R.K. Gupta and D.P. Agrawal, 2009. A novel algorithm for mining frequent itemsets from large database. Int. J. Inform. Technol. Knowl. Manage., 2: 223-229.

Toivonen, H., 1996. Sampling large databases for association rules. Proceedings of 22th International Conference on Very Large Databases, Sept. 3-6, Bombay, India, pp: 134-145.

Wang, J., J. Han and J. Pei, 2003a. CLOSET+: Searching for the best strategies for mining frequent closed itemsets. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 24-27, Washington, DC., pp: 236-245.

Wang, K., Y. Jiang and L. Lakshmanan, 2003b. Mining unexpected rules by pushing user dynamics. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery in Databases, Aug. 24-27, Washington, DC., pp: 246-255.

Xin, D., J. Han, X. Yan and H. Cheng, 2005. Mining compressed frequent-pattern sets. Proceedings of the 31st International Conference on Very Large Data Bases, Aug. 30-Sept. 2, Trondheim, Norway, pp: 709-720.

Xin, D., J. Han, Z. Shao and H. Liu, 2006. C-cubing: Efficient computation of closed cubes by aggregation-based checking. Proceedings of the 22nd International Conference on Data Engineering, April 3-7, Atlanta, Georgia, pp: 4-13.

Yan, X., J. Han R. Afshar, 2003. CloSpan: Mining closed sequential patterns in large datasets. Proc. SIAM Int. Conf. Data Min., 3: 166-177.

Yan, X., H. Cheng, J. Han and D. Xin, 2005a. Summarizing itemset patterns: A profile-based approach. Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Databases, Aug. 21-24, Chicago, IL., pp: 314-323.

Yan, X., P. Yu and J. Han, 2005b. Substructure similarity search in graph databases. Proceeding of the Special Interest Group on Management of Data, June 14-16, Baltimore, Maryland, pp: 766-777.

Yang, C., U. Fayyad and P.S. Bradley, 2001. Efficient discovery of error-tolerant frequent itemsets in high dimensions. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery in Databases, Aug. 26-29, San Fransisco, CA., pp: 194-203.

Yang, G., 2004. The complexity of mining maximal frequent itemsets andmaximal frequent patterns. Proceedings of the 10th ACM SIGKDD International Conference on Kowledge Discovery in Databases, Aug. 22-25, Seattle, WA., pp: 344-353.

Yu, J.X., Z. Chong, H. Lu and A. Zhou, 2004. False positive or false negative: Mining frequent itemsets from high speed transactional data streams. Proceedings of the 30th International Conference on Very Large Data Bases, Aug. 31-Sept. 3, Toronto, Canada, pp: 204-215.

Yun, U. and J. Leggett, 2005. Wfim: Weighted frequent itemsetminingwith a weight range and aminimum weight. Proceedings of the 2005 SIAM International Conference on Data Mining, April 21-23, Newport Beach, CA., pp: 636-640.

Zaiane, O.R., J. Han and H. Zhu, 2000. Mining recurrent items in multimedia with progressive resolution refinement. Proceedings of the 16th International Conference on Data Engineering, Feb. 29-March 3, San Diego, CA., USA., pp: 461-470.

Zaki, M.J., O. Mitsunori, S. Parthasarathy and L. Wei, 1996. Parallel data mining for association rules on shared memory multiprocessors. Proceedings of the 1996 ACM/IEEE Conference on High Performance Networking and Computing, Jan. 01, IEEE Computer Society, Washington, DC., USA., pp: 1-25.

Zaki, M.J., S. Parthasarathy, M. Ogihara and W. Li, 1997. Parallel algorithm for discovery of association rules. Data Min. Knowl. Discov., 1: 343-374.

Zaki, M.J., 1998. Efficient enumeration of frequent sequences. Proceedings of the 7th International Conference on Information and Knowledge Management, Nov. 2-7, Washington, DC., pp: 68-75.

Zaki, M.J., 2000. Scalable algorithms for association mining. IEEE Trans. Knowl. Data Eng., 12: 372-390.

Zaki, M.J., 2001. SPADE: An efficient algorithm for mining frequent sequences. Mach. Learn. J., 40: 31-60.

Zaki, M.J. and C.J. Hsiao, 2002. CHARM: An efficient algorithm for closed itemset mining. Proceedings of the 2nd SIAM International Conference on Data Mining, April 11-13, Arlington, VA., pp: 457-473.

Zhang, X., N. Mamoulis, D.W. Cheung and Y. Shou, 2004. Fast mining of spatial collocations. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery in Databases, Aug. 22-25, Seattle, WA., pp: 384-393.

Zaki, M.J., Member and H. Ching-Jui, 2005. Efficient algorithm for mining closed itemsets and their lattice structure. IEEE Trans. Knowledge Data Eng., 17: 462-478.

Zhu, F., X. Yan, J. Han, P.S. Yu and H. Cheng, 2007. Mining colossal frequent patterns by core pattern fusion. Proceedings of the 23rd International Conference on Data Engineering, April 15-21, Istanbul, Turkey, pp: 706-715.