Data Min Knowl Disc (2007) 15:55–86
DOI 10.1007/s10618-006-0059-1

Frequent pattern mining: current status and future directions

Jiawei Han · Hong Cheng · Dong Xin · Xifeng Yan

Received: 22 June 2006 / Accepted: 8 November 2006 / Published online: 27 January 2007

Abstract Frequent pattern mining has been a focused theme in data mining research for over a decade. Abundant literature has been dedicated to this research 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. In this article, we provide a brief overview of the current status of frequent pattern mining and discuss a few promising research directions. We believe that frequent pattern mining research has substantially broadened the scope of data analysis and will have deep impact on data mining methodologies and applications in the long run. However, there are still some challenging research issues that need to be solved before frequent pattern mining can claim a cornerstone approach in data mining applications.

Keywords Frequent pattern mining · Association rules · Data mining research · Applications

Responsible editor: Geoff Webb.

The work was supported in part by the U.S. National Science Foundation NSF IIS-05-13678/06-42771 and NSF BDI-05-15813. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

J. Han (✉)· H. Cheng · D. Xin · X. Yan
Department of Computer Science
University of Illinois, 1304 West Springfield Ave.
Urbana, IL 61801, USA
e-mail: hanj@cs.uiuc.edu

1 Introduction

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 data mining research.

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 habits by finding associations between the different items that customers place in their "shopping baskets". For instance, if customers are buying milk, how likely are they going to also buy cereal (and what kind of cereal) on the same trip to the supermarket? Such information can lead to increased sales by helping retailers do selective marketing and arrange their shelf space.

Since the first proposal of this new data mining task and its associated efficient mining algorithms, there have been hundreds of follow-up research publications, on various kinds of extensions and applications, ranging from scalable data mining methodologies, to handling a wide diversity of data types, various extended mining tasks, and a variety of new applications. With over a decade of substantial and fruitful research, it is time to perform an overview of this flourishing field and examine what more to be done in order to turn this technology a cornerstone approach in data mining applications.

In this article, we perform a high-level overview of frequent pattern mining methods, extensions and applications. With a rich body of literature on this theme, we organize our discussion into the following five themes: (1) efficient and scalable methods for mining frequent patterns, (2) mining interesting frequent patterns, (3) impact to data analysis and mining applications, (4) applications of frequent patterns, and (5) research directions. The remaining of the article is also organized in the corresponding five sections (Sects. 2 to 6), and we conclude our study in Sect. 7.

2 Efficient and scalable methods for mining frequent patterns

The concept of frequent itemset was first introduced for mining transaction databases (Agrawal et al. 1993). Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of all items. A k-itemset α, which consists of k items from I, is frequent if α occurs in a transaction database D no lower than $\theta|D|$ times, where θ is a user-specified

minimum support threshold (called *min_sup* in our text), and $|D|$ is the total number of transactions in D.

In Sect. 2.1, three basic frequent itemset mining methodologies: Apriori, FP-growth and Eclat, and their extensions, are introduced. In Sect. 2.2, multilevel, multidimensional, and quantitative association rule mining is discussed. In Sect. 2.3, the concepts of *closed* and *maximal* frequent itemsets and their related algorithms, are examined. Techniques for mining closed frequent itemsets from very high dimensional data sets and mining very long patterns (thus called colossal patterns) are presented in Sect. 2.4. Complex pattern mining such as sequential pattern mining and structural pattern mining is described in Sect. 2.5 and 2.6.

2.1 Basic mining methodologies: apriori, FP-growth and eclat

2.1.1 Apriori principle, apriori algorithm and its extensions

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., hashing technique (Park et al. 1995), partitioning technique (Savasere et al. 1995), sampling approach (Toivonen 1996), dynamic itemset counting (Brin et al. 1997), incremental mining (Cheung et al. 1996), parallel and distributed mining (Park et al. 1995; Agrawal and Shafer 1996; 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.

2.1.2 Mining frequent itemsets without candidate generation

In many cases, the Apriori algorithm significantly reduces the size of candidate sets using the Apriori principle. 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. (2000)

devised an FP-growth method that mines the complete set of frequent itemsets without candidate generation.

FP-growth works in a *divide-and-conquer* way. 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 "subdatabase", 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 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. (2001) 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).

2.1.3 Mining frequent itemsets using vertical data format

Both the Apriori and FP-growth methods mine frequent patterns 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 bought in transaction *TID*. Alternatively, mining can also be performed with data presented in *vertical data format* (*i.e.*, {*item*: *TID_set*}).

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. 2000). 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 $k \geq 1$). This is because the TID_set of each k-itemset carries the complete information required for counting such support.

Another related work which mines the frequent itemsets with the vertical data format is (Holsheimer et al. 1995). This work demonstrated that, though impressive results have been achieved for some data mining problems using highly specialized and clever data structures, one could also explore the potential of solving data mining problems using the general purpose database management systems (dbms).

2.2 Mining multilevel, multidimensional, and quantitative association rules

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 commonsense 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 (Srikant and Agrawal 1995; 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 (Srikant and Agrawal 1995). Efficient mining can also be derived if *min_sup* varies at different levels (Han and Kamber 2006). 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 binned intervals and then clustering mined quantitative association rules for concise representation as in the ARCS system, by Lent et al. (1997); based on x-monotone and rectilinear regions by Fukuda et al. (1996) and Yoda et al. (1997), or using distance-based (clustering) method over interval data, by Miller and Yang (1997). 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. (2004) 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.

2.3 Mining closed and maximal frequent itemsets

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.

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. 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 $\alpha \subset \beta$ 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. 2003), 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. The complexity of enumerating maximal itemsets is shown to be NP-hard. Ramesh

et al. (2003) characterized the length distribution of frequent and maximal frequent itemset collections. The conditions are also characterized under which one can embed such distributions in a database.

2.4 Mining high-dimensional datasets and mining colossal patterns

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. (2006) 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.

2.5 Mining sequential patterns

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 clickstreams, 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. We first introduce the preliminary concept about sequential patterns.

Let $I = \{i_1, i_2, \ldots, i_k\}$ be a set of all items. A subset of I is called an *itemset*. A *sequence* $\alpha = \langle t_1, t_2, \ldots, t_m \rangle$ $(t_i \subseteq I)$ is an ordered list. Each itemset in a sequence represents a set of events occurring at the same timestamp, while different itemsets occur at different times. For example, a customer shopping sequence could be buying several products on one trip to the store and making several subsequent purchases, e.g., buying a PC and some software tools, followed by buying a digital camera and a memory card, and finally buying a printer and some books.

Without loss of generality, we assume that the items in each itemset are sorted in certain order (such as alphabetic order). A sequence $\alpha = \langle a_1, a_2, \ldots, a_m \rangle$ is a *sub-sequence* of another sequence $\beta = \langle b_1, b_2, \ldots, b_n \rangle$, denoted by $\alpha \sqsubseteq \beta$ (if $\alpha \neq \beta$, written as $\alpha \sqsubset \beta$), if and only if $\exists i_1, i_2, \ldots, i_m$, such that $1 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n$ and $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \ldots,$ *and* $a_m \subseteq b_{i_m}$. We also call β a *super-sequence* of α, and β *contains* α. Given a sequence database $D = \{s_1, s_2, \ldots, s_n\}$, the *support* of a sequence α is the number of sequences in D which contain α. If the support of a sequence α satisfies a pre-specified *min_sup* threshold, α is a *frequent* sequential pattern.

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 $\langle itemset : (sequence\_ID, event\_ID) \rangle$. 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. (2001, 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 supersequence with the same support.

The CloSpan algorithm for mining closed sequential patterns was proposed by Yan et al. (2003). The method is based on a property of sequence databases, called *equivalence of projected databases*, stated as follows: *Two projected sequence databases,* $S|_\alpha = S|_\beta$, $\alpha \sqsubseteq \beta$, *are equivalent if and only if the total number of items in* $S|_\alpha$ *is equal to the total number of items in* $S|_\beta$, where $S|_\alpha$ is the projected database with respect to the prefix α. Based on this property, CloSpan can prune the non-closed sequences from further consideration during the mining process. A later algorithm called BIDE, a bidirectional search for mining frequent closed sequences was developed by Wang and Han (2004), which can further optimize this process by projecting sequence datasets in two directions.

The studies of sequential pattern mining have been extended in several different ways. Mannila et al. (1997) consider frequent episodes in sequences, where episodes are essentially acyclic graphs of events whose edges specify the temporal before-and-after relationship but without timing-interval restrictions. Sequence pattern mining for plan failures was proposed in Zaki et al. (1998). Garofalakis et al. (1999) proposed the use of regular expressions as a flexible constraint specification tool that enables user-controlled focus to be incorporated into the sequential pattern mining process. The embedding of multidimensional, multilevel information into a transformed sequence database for sequential pattern mining was proposed by Pinto et al. (2001). Pei et al. (2002) studied issues regarding constraint-based sequential pattern mining. CLUSEQ is a sequence clustering algorithm, developed by Yang and Wang (2003). An incremental sequential pattern mining algorithm, IncSpan, was proposed by Cheng et al. (2004). SeqIndex, efficient sequence indexing by frequent and discriminative analysis of sequential patterns, was studied by Cheng et al. (2005). A method for parallel mining of closed sequential patterns was proposed by Cong et al. (2005). A method, MSPX, for mining maximal sequential patterns by using multiple samples, was proposed by Luo and Chung (2005).

Data mining for periodicity analysis has been an interesting theme in data mining. Özden et al. (1998) studied methods for mining periodic or cyclic association rules. Lu et al. (1998) proposed intertransaction association rules, which are implication rules whose two sides are totally ordered episodes with tim-

ing-interval restrictions (on the events in the episodes and on the two sides). Bettini et al. (1998) consider a generalization of intertransaction association rules. The notion of mining partial periodicity was first proposed by Han, Dong, and Yin, together with a max-subpattern hit set method (Han et al. 1999). Ma and Hellerstein (2001) proposed a method for mining partially periodic event patterns with unknown periods. Yang et al. (2003) studied mining asynchronous periodic patterns in time-series data. Mining partial order from unordered 0-1 data was studied by Gionis et al. (2003) and Ukkonen et al. (2005). Pei et al. (2005) proposed an algorithm for mining frequent closed partial orders from string sequences.

2.6 Mining structural patterns: graphs, trees and lattices

Many scientific and commercial applications need patterns that are more complicated than frequent itemsets and sequential patterns. Such sophisticated patterns go beyond sets and sequences, toward trees, lattices, and graphs. As a general data structure, graphs have become increasingly important in modeling sophisticated structures and their interactions, with broad applications including chemical informatics, bioinformatics, computer vision, video indexing, text retrieval, and Web analysis.

Among the various kinds of graph patterns, *frequent substructures* are the very basic patterns that can be discovered in a collection of graphs. Recent studies have developed several frequent substructure mining methods. Washio and Motoda (2003) conducted a survey on graph-based data mining. Holder et al. (1994) proposed SUBDUE to do approximate substructure pattern discovery based on minimum description length and background knowledge. Dehaspe et al. (1998) applied inductive logic programming to predict chemical carcinogenicity by mining frequent substructures. Besides these studies, there are two basic approaches to the frequent substructure mining problem: an Apriori-based approach and a pattern-growth approach.

2.6.1 Apriori-based approach

Apriori-based frequent substructure mining algorithms share similar characteristics with Apriori-based frequent itemset mining algorithms. The search for frequent graphs starts with graphs of small "size", and proceeds in a bottom-up manner. At each iteration, the size of newly discovered frequent substructures is increased by one. These new substructures are first generated by joining two similar but slightly different frequent subgraphs that were discovered already. The frequency of the newly formed graphs is then checked.

Typical Apriori-based frequent substructure mining algorithms include AGM by Inokuchi et al. (1998), FSG by Kuramochi and Karypis (2001), and an edge-disjoint path-join algorithm by Vanetik et al. (2002).

The AGM algorithm uses a *vertex-based candidate generation* method that increases the substructure size by one vertex at each iteration. Two size-k frequent graphs are joined only when the two graphs have the same size-$(k-1)$

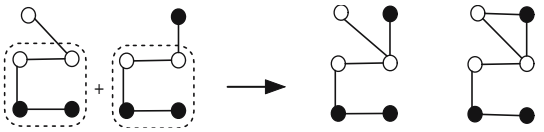

Fig. 1 Two substructures joined by two chains

subgraph. Here, *graph size* means the number of vertices in a graph. The newly formed candidate includes the common size-$(k-1)$ subgraph and the additional two vertices from the two size-k patterns. Because it is undetermined that whether there is an edge connecting the additional two vertices, AGM actually can form two candidates. Figure 1 depicts the two subgraphs joined by two chains.

The FSG algorithm adopts an *edge-based candidate generation* strategy that increases the substructure size by one edge in each iteration. Two size-k patterns are merged if and only if they share the same subgraph having k-1 edges. In the *edge-disjoint path* method, graphs are classified by the number of disjoint paths they have, and two paths are edge-disjoint if they do not share any common edge. A substructure pattern with k+1 disjoint paths is generated by joining substructures with k disjoint paths.

The Apriori-based algorithms mentioned above have considerable overhead when two size-k frequent substructures are joined to generate size-$(k+1)$ graph candidates. In order to avoid such overhead, non-Apriori-based algorithms have been developed, most of which adopt the pattern-growth methodology, as discussed below.

2.6.2 Pattern-growth approach

Pattern-growth-based graph pattern mining algorithms include gSpan by Yan and Han (2002), MoFa by Borgelt and Berthold (2002), FFSM by Huan et al. (2003), SPIN by Huan et al. (2004), and Gaston by Nijssen and Kok (2004). These algorithms are inspired by PrefixSpan (Pei et al. 2001), TreeMinerV (Zaki 2002), and FREQT (Asai et al. 2002) at mining sequences and trees, respectively.

The pattern-growth mining algorithm extends a frequent graph by adding a new edge, in every possible position. A potential problem with the edge extension is that the same graph can be discovered many times. The gSpan algorithm solves this problem by introducing a *right-most extension* technique, where the only extensions take place on the *right-most path*. A right-most path is the straight path from the starting vertex v_0 to the last vertex v_n, according to a depth-first search on the graph.

Besides the frequent substructure mining algorithms, constraint-based substructure mining algorithms have also been proposed. Mining closed graph patterns was studied by Yan and Han (2003) with the proposal of the algorithm, CloseGraph, as an extension of gSpan and CloSpan (Yan et al. 2003). Mining coherent subgraphs was studied by Huan et al. (2004). For mining relational graphs, Yan et al. (2005) proposed two algorithms, CloseCut and Splat, to discover exact dense frequent substructures in a set of relational graphs. For large-scale graph database mining, a disk-based frequent graph mining method was proposed by Wang et al. (2004). Jin et al. (2005) proposed an algo-

rithm, TSMiner, for mining frequent large-scale structures (defined as topological structures) from graph datasets. Techniques are developed for pushing constraints and specifying approximate matches. Kuramochi and Karypis (2004) proposed an algorithm, GREW, for finding patterns corresponding to connected subgraphs that have a large number of vertex-disjoint embeddings from a large graph. Ting and Bailey (2006) proposed an algorithm for mining the minimal contrast subgraph which is able to capture the structural differences between any two collections of graphs.

3 Mining interesting frequent patterns

Although numerous scalable methods have been developed for mining frequent patterns and closed (maximal) patterns, such mining often generates a huge number of frequent patterns. People would like to see or use only interesting ones. What are interesting patterns and how to mine them efficiently? To answer such questions, 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. These will be covered in this section.

3.1 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 several 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 $avg() \geq v$, can be performed by sorting items in each transaction in their value ascending or descending order for constrained pattern growth (Pei et al. 2001). 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. Constraint-based mining has also been explored in the context of sequential pattern mining (Garofalakis et al. 1999; Pei et al. 2002), as mentioned in Sect. 2.5.

3.2 Mining compressed or approximate patterns

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 and lossy compression, in terms of the information that the result set contains, compared with the whole set of frequent patterns.

Mining closed patterns, described in Sect . 2.3, is a lossless compression of frequent patterns. Mining all non-derivable frequent sets proposed by Calders and Goethals (2002) 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. (2006) 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. (2005), condensed pattern bases by Pei et al. (2002), k-summarized patterns or pattern profiles by Afrati et al. (2004) and Yan et al. (2005), and clustering-based compression by Xin et al. (2005).

For mining top-k most frequent closed patterns, a TFP algorithm (Wang et al. 2005) 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. (2005) 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. 2002) partitions patterns based on their support and then finds 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). A *representative pattern* can be selected for each cluster. Siebes et al. (2006) proposed a formulation with the MDL principle – the best set of frequent itemsets is the set that compresses the database best. Heuristic algorithms are developed for finding the subset of frequent itemsets that compresses the database.

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. A symmetric 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. (2006) developed a general model for mining approximate frequent itemsets (AFI) which controls errors of two directions in matrices formed by transactions and items.

3.3 From frequent patterns to interestingness and correlation analysis

Frequent itemset mining naturally 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 $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ be a set of items. An association rule takes the form of $\alpha \Rightarrow \beta$, where $\alpha \subset \mathcal{I}$, $\beta \subset \mathcal{I}$, and $\alpha \cap \beta = \phi$, 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 $\alpha \Rightarrow \beta[support, confidence, correlation]$. There are various correlation measures including *lift*, 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. (1997) 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 in Brin et al. (1997) 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. 2003) 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 (2004, 2005), 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).

4 Impact to data analysis and mining tasks

Frequent patterns discovered via mining processes not only themselves are interesting, but also useful to other data analysis and mining tasks, including (1) associative classification (Sect. 4.1), (2) clustering (Sect. 4.2), (3) cube computation and analysis (Sect. 4.3), and (4) gradient mining and multi-dimensional discriminant analysis (Sect.4.4).

4.1 Frequent pattern-based classification

Frequent itemsets have been demonstrated to be useful for classification, where association rules are generated and analyzed for use in classification (Liu et al. 1998; Dong and Li 1999; Li et al. 2000; Li et al. 2001; Yin and Han 2003; Cong et al. 2005;Wang and Karypis 2005). The general idea is that strong associations between frequent patterns and class labels can be discovered. Then the association rules are used for prediction. In many studies, associative classification has been found to be more accurate than some traditional classification methods, such as C4.5.

The CBA algorithm for associative classification was proposed by Liu et al. (1998). A classifier, using emerging patterns, was proposed by Dong and Li (1999) and Li et al. (2000). Classification based on Multiple Association Rules (CMAR) was presented in Li et al. (2001). Classification based on Predictive Association Rules (CPAR) was proposed in Yin and Han (2003).

A recent work on top-k rule mining proposed by Cong et al. (2005) discovers top-k covering rule groups for each row of gene expression profiles. It uses a *row enumeration* technique and introduces several pruning strategies to make the rule mining process very efficient. A classifier RCBT is constructed from the top-k covering rule groups. Prediction is based on a classification score which combines the support and confidence measures of the rules.

Another recent work, HARMONY, by Wang and Karypis (2005) is a rule-based classifier which directly mines the final set of classification rules. It uses an instance-centric rule-generation approach and assures for each training instance, one of the highest-confidence rules covering the instance is included in the final rule set. It also introduces several search and pruning strategies to make HARMONY more efficient and scalable than previous rule-based classifiers.

Cheng et al. (2007) conducted a systematic study and provided solid reasoning to support the methodology of frequent pattern-based classification. By building a connection between pattern frequency and discriminative measures, such as information gain and fisher score, it is shown that discriminative frequent patterns are essential for classification, whereas inclusion of infrequent patterns may not improve the classification accuracy due to their limited predictive power. A strategy is also proposed to set minimum support in frequent pattern mining for generating useful patterns. With this strategy, coupled with a proposed feature selection algorithm, discriminative frequent patterns can be generated for building high quality classifiers. Empirical studies demonstrate that the frequent pattern-based classification framework can achieve both high accuracy and good scalability in classifying large datasets.

In graph classification, Deshpande et al. (2003) used frequent subgraphs as features and built classification models based on them.

4.2 Frequent pattern-based cluster analysis

Cluster analysis in high-dimensional space is a challenging problem. Since it is easy to compute frequent patterns in subsets of high dimensions, it provides a promising direction for high-dimensional and subspace clustering.

For high-dimensional clustering, an Apriori-based dimension-growth subspace clustering algorithm called CLIQUE was proposed by Agrawal et al. (1998). It integrates density-based and grid-based clustering methods. The Apriori property is used to find clusterable subspaces and dense units are identified. The algorithm then finds adjacent dense grid units in the selected subspaces using a depth first search. Clusters are formed by combining these units using a greedy growth scheme.

An entropy-based subspace clustering algorithm for mining numerical data, called ENCLUS, was proposed by Cheng et al. (1999). ENCLUS uses the same Apriori property to mine interesting subspaces, as CLIQUE. However, this algorithm uses entropy as the basic measure. It is based on the observation that a subspace with clusters typically has lower entropy than a subspace without clusters.

Text mining based on key-words clustering and microarray data clustering are naturally high-dimensional clustering problem, and the frequent pattern-based approach starts to demonstrate its power and promise. Beil et al. (2002) proposed a method for frequent term-based text clustering. Wang et al. (2002) proposed pCluster, a pattern similarity-based clustering method for microarray data analysis, and demonstrated its effectiveness and efficiency for finding subspace clusters in high-dimensional space.

4.3 Frequent pattern analysis versus cube computation

Frequent pattern analysis and data cube computation share many similarities because both need to orderly compute a large number of itemsets. The former computes such itemsets by "joining" frequent k-itemsets for some lower k, whereas the latter computes such itemset by teaming up the corresponding items from the involving dimensions. A direct link of frequent pattern to data cube is the iceberg cube which consists of only the cells whose measure (usually count) is no lower than a user-specified threshold.

Algorithms for iceberg cube computation also share similar principles of frequent pattern mining. The first algorithm to compute iceberg cubes which also exploits the Apriori property is BUC (Beyer and Ramakrishnan 1999), proposed by Beyer and Ramakrishnan. The algorithm starts by reading the first dimension and partitioning it based on its distinct values. Then for each partition in the first dimension, it recursively computes the remaining dimensions until the size of the remaining partition is less than the minimum support. This bottom-up computation order facilitates Apriori-based pruning.

Several iceberg cube computation methods are derived from the FP-growth framework. The H-Cubing (Han et al. 2001) method, proposed by Han et al., uses a hyper-tree structure, called HTree, to facilitate cube computation, which can be viewed as an extension of FP-growth to (iceberg) cube computation.

The Star-Cubing algorithm (Xin et al. 2003), proposed by Xin et al., modifies the HTree structure and uses an integrated computation order to exploit computational share and iceberg pruning at the same time.

Similar to closed frequent pattern, closed cube was introduced as a lossless compression of a full data cube. A formal study of the closed representation of data cube was conducted by Lakshmanan et al. (2002). Condensed Cube (Wang et al. 2002) and Dwarf Cube (Sismanis et al. 2002) were also introduced to reduce the cube size by exploiting the closure semantics. The C-Cubing method (Xin et al. 2006), developed by Xin et al., facilitates the closed cube computation by a closeness measure and has shown its effectiveness.

4.4 Gradient mining and discriminant analysis

Many data analysis tasks can be viewed as searching or mining in a multidimensional space. Besides traditional multidimensional analysis and data mining tasks, one interesting task is to find notable changes and comparative differences. This leads to gradient mining and discriminant analysis.

Gradient mining is to mine notable changes in multidimensional space. An interesting notion, called *cubegrade* was introduced by Imielinski et al. (2002), which focuses on the notable changes in measures in the context of data cube by comparing a cube cell (referred as probe cell) with its gradient cells, namely, its ancestors, descendants, and siblings. A constrained gradient is a pair of probe cell and its gradient cell whose measures have notable difference. To mine the constrained gradients, an efficient algorithm, called LiveSet-Driven (Dong et al. 2004), was proposed by Dong et al. which finds all good gradient-probe cell pairs in one search pass. It utilizes measure-value analysis and dimension-match analysis in a set-oriented manner, to achieve bidirectional pruning between the sets of hopeful probe cells and of hopeful gradient cells.

Methods for discriminative analysis over sequence data have also been proposed. Ji et al. (2005) studied the mining of *minimal distinguishing subsequence* that occurs frequently in one class of sequences and infrequently in sequences of another class. Extracting strong and succinct contrast information between two sequential datasets can be useful in applications like protein comparison, document comparison, and construction of sequential classification models. An efficient algorithm, called ConSGapMiner, is developed in (Ji et al. 2005) to find all distinguishing subsequences by first generating candidates, then computing their frequency support, testing the gap satisfaction, and finally using post processing to remove all non-minimal answers.

5 Applications

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. With limited space, we are focused in this section only on a small number of successful applications: (1) indexing and similarity search of complex structured data (Sect. 5.1), (2) spatio-

temporal and multimedia data mining (Sect. 5.2), (3) stream data mining (Sect. 5.3), (4) web mining (Sect. 5.4), and (5) software bug mining and page-fetch prediction (Sect. 5.5).

5.1 Indexing and similarity search of complex structured data

Complex objects such as transaction sequence, event logs, proteins and images are widely used in many fields. Efficient search of these objects becomes a critical problem for many applications. Due to the large volume of data, it is inefficient to perform a sequential scan on the whole database and examine objects one by one. High performance indexing mechanisms thus are in heavy demand in filtering objects that obviously violate the query requirement.

gIndex (Yan et al. 2004) proposes a discriminative frequent pattern-based approach to index structures and graphs. A fundamental problem arises: if only frequent patterns are indexed, how to find those queries which only have infrequent patterns? gIndex (Yan et al. 2004) solved this problem by replacing the uniform support constraint with a size-increasing support function, which has very low support for small patterns but high support for large patterns. The concept developed in gIndex can also be applied to indexing sequences, trees, and other complicated structures as well. SeqIndex is one example using frequent pattern-based approach to index sequences. Taking frequent patterns as features, new strategies to perform structural similarity search were developed such as Grafil (Yan et al. 2005) and PIS (Yan et al. 2006).

5.2 Spatiotemporal and multimedia data mining

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). At the coarse level of resolution, one may use rough spatial approximation such as minimum bounding rectangles to estimate the frequent pattern candidates. Only those frequent pairs will need to be re-examined at finer levels of resolution using more refined but expensive spatial computation. Similar methodology has been used in mining co-location patterns, by Xiong et al. (2004) and Zhang et al. (2004), where fur-

ther optimization is performed by considering the spatial co-location property, *i.e.*, the spatially closely located objects usually have more interesting and closer relationships than the objects located far apart. Such optimization ideas can be extended to mining spatiotemporal sequential patterns as well, as shown in Cao et al. (2005). Moreover, 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.

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* is finding 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. Zaïane et al. (2000) takes those factors into consideration and developed a progressive refinement algorithm for mining multimedia associations.

5.3 Mining data streams

Tremendous and potentially infinite volumes of data streams are often generated by real-time surveillance systems, communication networks, Internet traffic, online transactions in the financial market or retail industry, electric power grids, industry production processes, scientific and engineering experiments, remote sensors, and other dynamic environments. 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., proposed a counting algorithm for finding frequent elements in data streams (Karp et al. 2003). 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 in 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. Chi et al. (2004) proposed an algorithm for mining closed frequent itemsets over a sliding window. A synopsis data structure is designed to monitor the transactions in the sliding window and a compact data structure — a closed enumeration tree is used to maintain a dynamically selected set of itemsets. Metwally et al. proposed a method for computing frequent and top-k elements in data streams by carefully using buffer space (Metwally et al. 2005). Jin and Agarwal (2005) proposed a one pass algorithm for frequent itemset mining which has deterministic bound on the accuracy and does not require any out-of-core summary structure. Lin et al. (2005) proposed an algorithm for mining frequent itemsets from data streams based on a time-sensitive sliding window.

5.4 Web mining

Web mining is the application of data mining techniques to discover patterns and knowledge from the Web (Kosala and Blockeel 2000; Srivastava et al. 2000). 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 clickthrough 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; Punin and Krishnamoorthy 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; Nanopoulos and Manolopoulos 2001).

5.5 Software bug mining and system caching

Analyzing 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. Liu et al. (2005) developed a novel method to classify the structured traces of program executions using software behavior graphs. The classification framework is built on an integration of frequent graph mining and SVM classification. Suspicious buggy regions are identified through the capture of the classification accuracy change, which is measured incrementally during program execution.

CP-Miner (Li et al. 2004), relying on pattern mining techniques, is able to identify copy-pasted code for bug isolation. 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. (2004), was proposed to mine frequent sequential patterns of correlated blocks from block access traces. It is built based on the observation that block correlations are common semantic patterns in storage systems. These correlations can be exploited for improving the effectiveness of storage caching, prefetching, data layout, and disk scheduling. In addition to storage system, frequent XML query patterns are also used to improve the caching performance of XML management systems (Yang et al. 2003).

6 Research directions

With abundant literature published in research into frequent pattern mining, one may wonder whether we have solved most of the critical problems related to frequent pattern mining so that the solutions provided are good enough for most of the data mining tasks. However, based on our view, there are still several critical research problems that need to be solved before frequent pattern mining can become a cornerstone approach in data mining applications.

First, the most focused and extensively studied topic in frequent pattern mining is perhaps scalable mining methods. Have we exhausted our search for efficient mining methodologies so that one can readily derive desired pattern sets with satisfactory performance? The answer, to many's surprise, is probably negative. We feel the bottleneck of frequent pattern mining is not on whether we can derive the *complete* set of frequent patterns under certain constraints efficiently but on whether we can derive a *compact but high quality* set of patterns that are most useful in applications. The set of frequent patterns derived by most of the current pattern mining methods is too huge for effective usage. There are proposals on reduction of such a huge set, including closed patterns, maximal patterns, approximate patterns, condensed pattern bases, representative patterns, clustered patterns, and discriminative frequent patterns, as introduced in the previous sections. However, it is still not clear what kind of patterns will give us satisfactory pattern sets in both compactness and representative quality for a particular application, and whether we can mine such patterns directly and efficiently. Much research is still needed to substantially reduce the size of derived pattern sets and enhance the quality of retained patterns.

Second, although we have efficient methods for mining precise and complete set of frequent patterns, approximate frequent patterns could be the best choice in many applications. For example, in the analysis of DNA or protein sequences, one would like to find long sequence patterns that approximately match the sequences in biological entities, similar to BLAST. Can we mine such patterns effectively? Much research is still needed to make such mining more effective than the currently available tools in bioinformatics.

Third, to make frequent pattern mining an essential task in data mining, much research is needed to further develop pattern-based mining methods. For example, classification is an essential task in data mining. Can we construct better classification models using frequent patterns than most other classification methods? What kind of frequent patterns are more effective than other frequent patterns. Can we mine such pattern directly from data? These questions need to be answered before frequent patterns can play an essential role in several major data mining tasks, such as classification.

Fourth, we need mechanisms for deep understanding and interpretation of patterns, e.g., semantic annotation for frequent patterns, and contextual analysis of frequent patterns. The main research work on pattern analysis has been focused on pattern composition (e.g., the set of items in item-set patterns) and frequency. The semantic of a frequent pattern includes deeper information: what is the meaning of the pattern; what are the synonym patterns; and what are the typical transactions that this pattern resides? In many cases, frequent patterns are mined from certain data sets which also contain structural information. For example, the shopping transaction data is normally tagged with time and location. Some text data (e.g., research papers) has associated attributes (e.g., authors, journals, or conferences). A contextual analysis of frequent patterns over the structural information can help respond questions like "why this pattern is frequent?" An example answer could be "this pattern is frequent because it happens heavily during time T_1 to T_2". We believe the deep understanding of frequent patterns is essential to improve the interpretability and the usability of frequent patterns. One initial study in this direction is done by Mei et al. (2006).

Finally, applications often raise new research issues and bring deep insight on the strength and weakness of an existing solution. This is also true for frequent pattern mining. On one side, it is important to go to the core part of pattern mining algorithms, and analyze the theoretical properties of different solutions. On the other side, although we only cover a small subset of applications in this article, frequent pattern mining has claimed a broad spectrum of applications and demonstrated its strength at solving some problems. Much work is needed to explore new applications of frequent pattern mining. For example, bioinformatics has raised a lot of challenging problems, and we believe frequent pattern mining may contribute a good deal to it with further research efforts.

7 Conclusions

In this article, we present a brief overview of the current status and future directions of frequent pattern mining. With over a decade of extensive research, there

have been hundreds of research publications and tremendous research, development and application activities in this domain. It is impossible for us to give a complete coverage on this topic with limited space and our limited knowledge. Hopefully, this short overview may provide a rough outline of the recent work and give people a general view of the field. In general, we feel that as a young research field in data mining, frequent pattern mining has achieved tremendous progress and claimed a good set of applications. However, in-depth research is still needed on several critical issues so that the field may have its long lasting and deep impact in data mining applications.

References

Afrati FN, Gionis A, Mannila H (2004) Approximating a collection of frequent sets. In: Proceedings of the 2004 ACM SIGKDD international conference knowledge discovery in databases (KDD'04), Seattle, WA, pp 12–19

Agarwal R, Aggarwal CC, Prasad VVV (2001) A tree projection algorithm for generation of frequent itemsets. J Parallel Distribut Comput 61:350–371

Aggarwal CC, Yu PS (1998) A new framework for itemset generation. In: Proceedings of the 1998 ACM symposium on principles of database systems (PODS'98), Seattle, WA, pp 18–24

Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM-SIGMOD international conference on management of data (SIGMOD'98), Seattle, WA, pp 94–105

Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM-SIGMOD international conference on management of data (SIGMOD'93), Washington, DC, pp 207–216

Agrawal R, Shafer JC (1996) Parallel mining of association rules: design, implementation, and experience. IEEE Trans Knowl Data Eng 8:962–969

Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 1994 international conference on very large data bases (VLDB'94), Santiago, Chile, pp 487–499

Agrawal R, Srikant R (1995) Mining sequential patterns. In: Proceedings of the 1995 international conference on data engineering (ICDE'95), Taipei, Taiwan, pp 3–14

Ahmed KM, El-Makky NM, Taha Y (2000) A note on "beyond market basket: generalizing association rules to correlations". SIGKDD Explorations 1:46–48

Asai T, Abe K, Kawasoe S, Arimura H, Satamoto H, Arikawa S (2002) Efficient substructure discovery from large semi-structured data. In: Proceedings of the 2002 SIAM international conference on data mining (SDM'02), Arlington, VA, pp 158–174

Aumann Y, Lindell Y (1999) A statistical theory for quantitative association rules. In: Proceeding of the 1999 international conference on knowledge discovery and data mining (KDD'99), San Diego, CA, pp 261–270

Bayardo RJ (1998) Efficiently mining long patterns from databases. In: Proceeding of the 1998 ACM-SIGMOD international conference on management of data (SIGMOD'98), Seattle, WA, pp 85–93

Beil F, Ester M, Xu X (2002) Frequent term-based text clustering. In: Proceeding of the 2002 ACM SIGKDD international conference on knowledge discovery in databases (KDD'02), Edmonton, Canada, pp 436–442

Bettini C, Sean Wang X, Jajodia S (1998) Mining temporal relationships with multiple granularities in time sequences. Bull Tech Committee Data Eng 21:32–38

Beyer K, Ramakrishnan R (1999) Bottom-up computation of sparse and iceberg cubes. In: Proceeding of the 1999 ACM-SIGMOD international conference on management of data (SIGMOD'99), Philadelphia, PA, pp 359–370

Blanchard J, Guillet F, Gras R, Briand H (2005) Using information-theoretic measures to assess association rule interestingness. In: Proceeding of the 2005 international conference on data mining (ICDM'05), Houston, TX, pp 66–73

Bonchi F, Giannotti F, Mazzanti A, Pedreschi D (2003) Exante: anticipated data reduction in constrained pattern mining. In: Proceeding of the 7th European conference on principles and pratice of knowledge discovery in databases (PKDD'03), pp 59–70

Bonchi F, Lucchese C (2004) On closed constrained frequent pattern mining. In: Proceeding of the 2004 international conference on data mining (ICDM'04), Brighton, UK, pp 35–42

Borgelt C, Berthold MR (2002) Mining molecular fragments: finding relevant substructures of molecules. In: Proceeding of the 2002 international conference on data mining (ICDM'02), Maebashi, Japan, pp 211–218

Brin S, Motwani R, Silverstein C (1997) Beyond market basket: generalizing association rules to correlations. In: Proceeding of the 1997 ACM-SIGMOD international conference on management of data (SIGMOD'97), Tucson, AZ, pp 265–276

Brin S, Motwani R, Ullman JD, Tsur S (1997) Dynamic itemset counting and implication rules for market basket analysis. In: Proceeding of the 1997 ACM-SIGMOD international conference on management of data (SIGMOD'97), Tucson, AZ, pp 255–264

Bucila C, Gehrke J, Kifer D, White W (2003) DualMiner: a dual-pruning algorithm for itemsets with constraints. Data Min knowl discov 7:241–272

Burdick D, Calimlim M, Gehrke J (2001) MAFIA: a maximal frequent itemset algorithm for transactional databases. In: Proceeding of the 2001 international conference on data engineering (ICDE'01), Heidelberg, Germany, pp 443–452

Calders T, Goethals B (2002) Mining all non-derivable frequent itemsets. In: Proceeding of the 2002 European conference on principles and pratice of knowledge discovery in databases (PKDD'02), Helsinki, Finland, pp 74–85

Calders T, Goethals B (2005) Depth-first non-derivable itemset mining. In: Proceeding of the 2005 SIAM international conference on data mining (SDM'05), Newport Beach, CA, pp 250–261

Cao H, Mamoulis N, Cheung DW (2005) Mining frequent spatio-temporal sequential patterns. In: Proceeding of the 2005 international conference on data mining (ICDM'05), Houston, TX, pp 82–89

Chang J, Lee W (2003) Finding recent frequent itemsets adaptively over online data streams. In: Proceeding of the 2003 international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 487–492

Chen MS, Park JS, Yu PS (1996) Data mining for path traversal patterns in a web environment. In: Proceeding of the 16th international conference on distributed computing systems, pp 385–392

Cheng CH, Fu AW, Zhang Y (1999) Entropy-based subspace clustering for mining numerical data. In: Proceeding of the 1999 international conference on knowledge discovery and data mining (KDD'99), San Diego, CA, pp 84–93

Cheng H, Yan X, Han J (2004) IncSpan: incremental mining of sequential patterns in large In: Proceeding of the 2004 ACM SIGKDD international conference on knowledge discovery in databases (KDD'04), Seattle, WA, pp 527–532

Cheng H, Yan X, Han J (2005) Seqindex: indexing sequences by sequential pattern analysis. In: Proceeding of the 2005 SIAM international conference on data mining (SDM'05), Newport Beach, CA, pp 601–605

Cheng H, Yan X, Han J, Hsu C (2007) Discriminative frequent pattern analysis for effective classification. In: Proceeding of the 2007 international conference on data engineering (ICDE'07), Istanbul, Turkey

Cheung DW, Han J, Ng V, Fu A, Fu Y (1996) A fast distributed algorithm for mining association rules. In: Proceeding of the 1996 international conference on parallel and distributed information systems, Miami Beach, FL, pp 31–44

Cheung DW, Han J, Ng V, Wong CY (1996) Maintenance of discovered association rules in large an incremental updating technique. In: Proceeding of the 1996 international conference on data engineering (ICDE'96), New Orleans, LA, pp 106–114

Chi Y, Wang H, Yu PS, Muntz R (2004) Moment: maintaining closed frequent itemsets over a stream sliding window. In: Proceeding of the 2004 international conference on data mining (ICDM'04), Brighton, UK, pp 59–66

Cong S, Han J, Padua D (2005) Parallel mining of closed sequential patterns. In: Proceeding of the 2005 ACM SIGKDD international conference on knowledge discovery in databases (KDD'05), Chicago, IL, pp 562–567

Cong G, Tan K-L, Tung AKH, Xu X (2005) Mining top-k covering rule groups for gene expression data. In: Proceeding of the 2005 ACM-SIGMOD international conference on management of data (SIGMOD'05), Baltimore, MD, pp 670–681

Deshpande M, Kuramochi M, Karypis G (2003) Frequent sub-structure-based approaches for classifying chemical compounds. In: Proceeding of the 2002 international conference on data mining (ICDM'03), Melbourne, FL, pp 35–42

Dong G, Han J, Lam J, Pei J, Wang K, Zou W (2004) Mining constrained gradients in multi-dimensional databases. IEEE Trans Knowl Data Eng 16:922–938

Dehaspe L, Toivonen H, King R (1998) Finding frequent substructures in chemical compounds. In: Proceeding of the 1998 international conference on knowledge discovery and data mining (KDD'98), New York, NY, pp 30–36

Dong G, Li J (1999) Efficient mining of emerging patterns: discovering trends and differences. In: Proceeding of the 1999 international conference on knowledge discovery and data mining (KDD'99), San Diego, CA, pp 43–52

Eirinaki M, Vazirgiannis M (2003) Web mining for web personalization. ACM Trans Inter Tech 3:1–27

Fukuda T, Morimoto Y, Morishita S, Tokuyama T (1996) Data mining using two-dimensional optimized association rules: scheme, algorithms, and visualization. In: Proceeding of the 1996 ACM-SIGMOD international conference management of data (SIGMOD'96), Montreal, Canada, pp 13–23

Gade K, Wang J, Karypis G (2004) Efficient closed pattern mining in the presence of tough block constraints. In: Proceeding of the 2004 international conference on knowledge discovery and data mining (KDD'04), Seattle, WA, pp 138–147

Garofalakis M, Rastogi R, Shim K (1999) SPIRIT: Sequential pattern mining with regular expression constraints. In: Proceeding of the 1999 international conference on Very large data bases (VLDB'99), Edinburgh, UK, pp 223–234

Geerts F, Goethals B, Bussche J (2001) A tight upper bound on the number of candidate patterns. In: Proceeding of the 2001 international conference on data mining (ICDM'01), San Jose, CA, pp 155–162

Gionis A, Kujala T, Mannila H (2003) Fragments of order. In: Proceeding of the 2003 international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 129–136

Gionis A, Mannila H, Mielikäinen T, Tsaparas P (2006) Assessing data mining results via swap randomization. In: Proceeding of the 2006 ACM SIGKDD international conference on knowledge discovery in databases (KDD'06), Philadelphia, PA, pp 167–176

Goethals B, Zaki M (2003) An introduction to workshop on frequent itemset mining implementations. In: Proceeding of the ICDM'03 international workshop on frequent itemset mining implementations (FIMI'03), Melbourne, FL, pp 1–13

Grahne G, Lakshmanan L, Wang X (2000) Efficient mining of constrained correlated sets. In: Proceeding of the 2000 international conference on data engineering (ICDE'00), San Diego, CA, pp 512–521

Grahne G, Zhu J (2003)Efficiently using prefix-trees in mining frequent itemsets. In: Proceeding of the ICDM'03 international workshop on frequent itemset mining implementations (FIMI'03), Melbourne, FL, pp 123–132

Han J, Dong G, Yin Y (1999) Efficient mining of partial periodic patterns in time series database. In: Proceeding of the 1999 international conference on data engineering (ICDE'99), Sydney, Australia, pp 106–115

Han J, Fu Y (1995) Discovery of multiple-level association rules from large databases. In: Proceeding of the 1995 international conference on very large data bases (VLDB'95), Zurich, Switzerland, pp 420–431

Han J, Kamber M (2006) Data mining: concepts and techniques, 2nd edn. Morgan Kaufmann

Han J, Pei J, Dong G, Wang K (2001) Efficient computation of iceberg cubes with complex measures. In: Proceeding of the 2001 ACM-SIGMOD international conference on management of data (SIGMOD'01), Santa Barbara, CA, pp 1–12

Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD'00), Dallas, TX, pp 1–12

Hilderman RJ, Hamilton HJ (2001) Knowledge discovery and measures of interest. Kluwer Academic

Holder LB, Cook DJ, Djoko S (1994) Substructure discovery in the subdue system. In: Proceeding of the AAAI'94 workshop knowledge discovery in databases (KDD'94), Seattle, WA, pp 169–180
Holsheimer M, Kersten M, Mannila H, Toivonen H (1995) A perspective on databases and data mining. In Proceeding of the 1995 international conference on knowledge discovery and data mining (KDD'95), Montreal, Canada, pp 150–155
Huan J, Wang W, Bandyopadhyay D, Snoeyink J, Prins J, Tropsha A (2004) Mining spatial motifs from protein structure graphs. In: Proceeding of the 8th international conference on research in computational molecular biology (RECOMB), San Diego, CA, pp 308–315
Huan J, Wang W, Prins J (2003) Efficient mining of frequent subgraph in the presence of isomorphism. In: Proceeding of the 2003 international conference on data mining (ICDM'03), Melbourne, FL, pp 549–552
Huan J, Wang W, Prins J, Yang J (2004) Spin: mining maximal frequent subgraphs from graph databases. In: Proceeding of the 2004 ACM SIGKDD international conference on knowledge discovery in databases (KDD'04), Seattle, WA, pp 581–586
Imielinski T, Khachiyan L, Abdulghani A (2002) Cubegrades: generalizing association rules. Data Min Knowl Discov 6:219–258
Inokuchi A, Washio T, Motoda H (2000) An apriori-based algorithm for mining frequent substructures from graph data. In: Proceeding of the 2000 European symposium on the principle of data mining and knowledge discovery (PKDD'00), Lyon, France, pp 13–23
Jaroszewicz S, Scheffer T (2005) Fast discovery of unexpected patterns in data relative to a bayesian network. In: Proceeding of the 2005 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'05), Chicago, IL, pp 118–127
Jaroszewicz S, Simovici D (2004) interestingness of frequent itemsets using bayesian networks as background knowledge. In: Proceeding of the 2004 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'04), Seattle, WA, pp 178–186
Ji X, Bailey J, Dong G (2005) Mining minimal distinguishing subsequence patterns with gap constraints. In: Proceeding of the 2005 international conference on data mining (ICDM'05), Houston, TX, pp 194–201
Jin R, Agrawal G (2005) An algorithm for in-core frequent itemset mining on streaming data. In: Proceeding of the 2005 international conference on data mining (ICDM'05), Houston, TX, pp 210–217
Jin R, Wang C, Polshakov D, Parthasarathy S, Agrawal G (2005) Discovering frequent topological structures from graph datasets. In: Proceeding of the 2005 ACM SIGKDD international conference on knowledge discovery in databases (KDD'05), Chicago, IL, pp 606–611
Kamber M, Han J, Chiang JY (1997) Metarule-guided mining of multi-dimensional association rules using data cubes. In: Proceeding of the 1997 international conference on knowledge discovery and data mining (KDD'97), Newport Beach, CA, pp 207–210
Karp RM, Papadimitriou CH, Shenker S (2003) A simple algorithm for finding frequent elements in streams and bags. ACM Trans Database Syst, 28:51–55
Koperski K, Han J (1995) Discovery of spatial association rules in geographic information databases. In: Proceeding of the 1995 international symposium on large spatial databases (SSD'95), Portland, ME, pp 47–66
Kosala R, Blockeel H (2000) Web mining research: a survey. SIGKDD Explor 2
Kuramochi M, Karypis G (2001) Frequent subgraph discovery. In: Proceeding of the 2001 international conference on data mining (ICDM'01), San Jose, CA, pp 313–320
Kuramochi M, Karypis G (2004) GREW: a scalable frequent subgraph discovery algorithm. In Proceeding of the 2004 international conference on data mining (ICDM'04), Brighton, UK, pp 439–442
Lakshmanan LVS, Ng R, Han J, Pang A (1999) Optimization of constrained frequent set queries with 2-variable constraints. In: Proceeding of the 1999 ACM-SIGMOD international conference on management of data (SIGMOD'99), Philadelphia, PA, pp 157–168
Lakshmanan LVS, Pei J, Han J (2002) Quotient cube: how to summarize the semantics of a data cube. In: Proceeding of the 2002 international conference on very large data bases (VLDB'02), Hong Kong, China, pp 778–789
Lee Y-K, Kim W-Y, Cai YD, Han J (2003) CoMine: efficient mining of correlated patterns. In: Proceeding of the 2003 international conference on data mining (ICDM'03), Melbourne, FL, pp 581–584

Lent B, Swami A, Widom J (1997) Clustering association rules. In: Proceeding of the 1997 international conference on data engineering (ICDE'97), Birmingham, England, pp 220–231
Li Z, Chen Z, Srinivasan SM, Zhou Y (2004) C-Miner: mining block correlations in storage systems. In: Proceeding of the 2004 USENIX conference on file and storage technologies (FAST'04), San Francisco, CA, pp 173–186
Li J, Dong G, Ramamohanrarao K (2000) Making use of the most expressive jumping emerging patterns for classification. In: Proceeding of the 2000 Pacific-Asia conference on knowledge discovery and data mining (PAKDD'00), Kyoto, Japan, pp 220–232
Li X, Han J, Kim S (2006) Motion-alert: automatic anomaly detection in massive moving objects. In: IEEE international conference on intelligence and security informatics (ISI'06), San Diego, CA, pp 166–177
Li W, Han J, Pei J (2001) CMAR: accurate and efficient classification based on multiple class-association rules. In: Proceeding of the 2001 international conference on data mining (ICDM'01), San Jose, CA, pp 369–376
Li Z, Lu S, Myagmar S, Zhou Y (2004) CP-Miner: a tool for finding copy-paste and related bugs in operating system code. In: Proceeding of the 2004 symposium on operating systems design and implementation (OSDI'04), San Francisco, CA, pp 289–302
Li Z, Zhou Y (2005) PR-Miner: Automatically extracting implicit programming rules and detecting violations in large software code. In: Proceeding of the 2005 ACM SIGSOFT symposium on foundations software eng (FSE'05), Lisbon, Portugal, pp 306–315
Lin C, Chiu D, Wu Y, Chen A (2005) Mining frequent itemsets from data streams with a time-sensitive sliding window. In: Proceeding of the 2005 SIAM international conference on data mining (SDM'05), Newport Beach, pp 68–79
Liu H, Han J, Xin D, Shao Z (2006) Mining frequent patterns on very high dimensional data: a top-down row enumeration approach. In: Proceeding of the 2006 SIAM international conference on data mining (SDM'06), Bethesda, MD, pp 280–291
Liu B, Hsu W, Ma Y (1998) Integrating classification and association rule mining. In: Proceeding of the 1998 international conference on knowledge discovery and data mining (KDD'98), New York, NY, pp 80–86
Liu G, Li J, Wong L, Hsu W (2006) Positive borders or negative borders: how to make lossless generator based representations concise. In: Proceeding of the 2006 SIAM international conference on data mining (SDM'06), Bethesda, MD, pp 467–471
Liu G, Lu H, Lou W, Yu JX (2003) On computing, storing and querying frequent patterns. In: Proceeding of the 2003 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 607–612
Liu J, Paulsen S, Sun X, Wang W, Nobel A, Prins J (2006) Mining approximate frequent itemsets in the presence of noise: algorithm and analysis. In: Proceeding of the 2006 SIAM international conference on data mining (SDM'06), Bethesda, MD, pp 405–416
Liu J, Pan Y, Wang K, Han J (2002) Mining frequent item sets by opportunistic projection. In: Proceeding of the 2002 ACM SIGKDD international conference on knowledge discovery in databases (KDD'02), Edmonton, Canada, pp 239–248
Liu C, Yan X, Yu H, Han J, Yu PS (2005) Mining behavior graphs for "backtrace" of noncrashing bugs. In: Proceeding of the 2005 SIAM international conference on data mining (SDM'05), Newport Beach, pp 286–297
Lu H, Han J, Feng L (1998) Stock movement and n-dimensional inter-transaction association rules. In: Proceeding of the 1998 SIGMOD workshop research issues on data mining and knowledge discovery (DMKD'98), Seattle, WA, pp 12:1–12:7
Luo C, Chung S (2005) Efficient mining of maximal sequential patterns using multiple samples. In: Proceeding of the 2005 SIAM international conference on data mining (SDM'05), Newport Beach, CA, pp 415–426
Ma S, Hellerstein JL (2001) Mining partially periodic event patterns with unknown periods. In: Proceeding of the 2001 international conference on data engineering (ICDE'01), Heidelberg, Germany, pp 205–214
Manku G, Motwani R (2002) Approximate frequency counts over data streams. In: Proceeding of the 2002 international conference on very large data bases (VLDB'02), Hong Kong, China, pp 346–357

Mannila H, Toivonen H, Verkamo AI (1994) Efficient algorithms for discovering association rules. In: Proceeding of the AAAI'94 workshop knowledge discovery in databases (KDD'94), Seattle, WA, pp 181–192

Mannila H, Toivonen H, Verkamo AI (1997) Discovery of frequent episodes in event sequences. Data Min Knowl Discov 1:259–289

Mei Q, Xin D, Cheng H, Han J, Zhai C (2006) Generating semantic annotations for frequent patterns with context analysis. In: Proceeding of the 2006 ACM SIGKDD international conference on knowledge discovery in databases (KDD'06), Philadelphia, PA, pp 337–346

Metwally A, Agrawal D, El Abbadi A (2005) Efficient computation of frequent and top-k elements in data streams. In: Proceeding of the 2005 international conference on database theory (ICDT'05), Edinburgh, UK, pp 398–412

Miller RJ, Yang Y (1997) Association rules over interval data. In: Proceeding of the 1997 ACM-SIGMOD international conference on management of data (SIGMOD'97), Tucson, AZ, pp 452–461

Nanopoulos A, Manolopoulos Y (2001) Mining patterns from graph traversals. Data Knowl Eng 37:243–266

Ng R, Lakshmanan LVS, Han J, Pang A (1998) Exploratory mining and pruning optimizations of constrained associations rules. In: Proceeding of the 1998 ACM-SIGMOD international conference on management of data (SIGMOD'98), Seattle, WA, pp 13–24

Nijssen S, Kok J (2004) A quickstart in frequent structure mining can make a difference. In: Proceeding of the 2004 ACM SIGKDD international conference on kowledge discovery in databases (KDD'04), Seattle, WA, pp 647–652

Omiecinski E (2003) Alternative interest measures for mining associations. IEEE Trans Knowl and data engineering, 15:57–69

Özden B, Ramaswamy S, Silberschatz A (1998) Cyclic association rules. In: Proceeding of the 1998 international conference on data engineering (ICDE'98), Orlando, FL, pp 412–421

Pan F, Cong G, Tung AKH, Yang J, Zaki M (2003) CARPENTER: finding closed patterns in long biological datasets. In: Proceeding of the 2003 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 637–642

Pan F, Tung AKH, Cong G, Xu X (2004) COBBLER: combining column, and row enumeration for closed pattern discovery. In: Proceeding of the 2004 international conference on scientific and statistical database management (SSDBM'04), Santorini Island, Greece, pp 21–30

Park JS, Chen MS, Yu PS (1995) An effective hash-based algorithm for mining association rules. In: Proceeding of the 1995 ACM-SIGMOD international conference on management of data (SIGMOD'95), San Jose, CA, pp 175–186

Park JS, Chen MS, Yu PS (1995) Efficient parallel mining for association rules. In: Proceeding of the 4th international conference on information and knowledge management, Baltimore, MD, pp 31–36

Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Discovering frequent closed itemsets for association rules. In: Proceeding of the 7th international conference on database theory (ICDT'99), Jerusalem, Israel, pp 398–416

Pei J, Dong G, Zou W, Han J (2002) On computing condensed frequent pattern bases. In: Proceeding of the 2002 international conference on data mining (ICDM'02), Maebashi, Japan, pp 378–385

Pei J, Han J, Lakshmanan LVS (2001) Mining frequent itemsets with convertible constraints. In: Proceeding of the 2001 international conference on data engineering (ICDE'01), Heidelberg, Germany, pp 433–332

Pei J, Han J, Mao R (2000) CLOSET: an efficient algorithm for mining frequent closed itemsets. In: Proceeding of the 2000 ACM-SIGMOD international workshop data mining and knowledge discovery (DMKD'00), Dallas, TX, pp 11–20

Pei J, Han J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, Hsu M-C (2001) PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: Proceeding of the 2001 international conference on data engineering (ICDE'01), Heidelberg, Germany, pp 215–224

Pei J, Han J, Mortazavi-Asl B, Wang J, Pinto H, Chen Q, Dayal U, Hsu M-C (2004) Mining sequential patterns by pattern-growth: the prefixspan approach. IEEE Trans Knowl Data Eng 16:1424–1440

Pei J, Han J, Mortazavi-Asl B, Zhu H (2000) Mining access patterns efficiently from web logs. In: Proceeding of the 2000 Pacific-Asia conference on knowledge discovery and data mining (PAKDD'00), Kyoto, Japan, pp 396–407

Pei J, Han J, Wang W (2002) Constraint-based sequential pattern mining in large databases. In: Proceeding of the 2002 international conference on information and knowledge management (CIKM'02), McLean, VA, pp 18–25

Pei J, Liu J, Wang H, Wang K, Yu PS, Yang J (2005) Efficiently mining frequent closed partial orders. In: Proceeding of the 2005 international conference on data mining (ICDM'05), Houston, TX, pp 753–756

Piatetsky-Shapiro G (1991) Notes of AAAI'91 workshop knowledge discovery in databases (KDD'91). AAAI/MIT Press, Anaheim, CA

Pinto H, Han J, Pei J, Wang K, Chen Q, Dayal U (2001) Multi-dimensional sequential pattern mining. In: Proceeding of the 2001 international conference on information and knowledge management (CIKM'01), Atlanta, GA, pp 81–88

Punin J, Krishnamoorthy M, Zaki M (2001) Web usage mining: languages and algorithms. Springer-Verlag

Ramesh G, Maniatty WA, Zaki MJ (2003) Feasible itemset distributions in data mining: theory and application. In: Proceeding of the 2003 ACM symposium on principles of database systems (PODS'03), San Diego, CA, pp 284–295

Sarawagi S, Thomas S, Agrawal R (1998) Integrating association rule mining with relational database systems: alternatives and implications. In: Proceeding of the 1998 ACM-SIGMOD international conference on management of data (SIGMOD'98), Seattle, WA, pp 343–354

Savasere A, Omiecinski E, Navathe S (1995) An efficient algorithm for mining association rules in large databases. In: Proceeding of the 1995 international conference on very large data bases (VLDB'95), Zurich, Switzerland, pp 432–443

Seppänen J, Mannila H (2004) Dense itemsets. In: Proceeding of the 2004 international conference on knowledge discovery and data mining (KDD'04), Seattle, WA, pp 683–688

Shekar B, Natarajan R (2004) A transaction-based neighbourhood-driven approach to quantifying interestingness of assoication rules. In: Proceeding of the 2004 international conference on data mining (ICDM'04), Brighton, UK, pp 194–201

Siebes A, Vreeken J, Leeuwen M (2006) Item sets that compress. In: Proceeding of the 2006 SIAM international conference on data mining (SDM'06), Bethesda, MD, pp 393–404

Silverstein C, Brin S, Motwani R, Ullman JD (1998) Scalable techniques for mining causal structures. In: Proceeding of the 1998 international conference on very large data bases (VLDB'98), New York, NY, pp 594–605

Sismanis Y, Roussopoulos N, Deligianannakis A, Kotidis Y (2002) Dwarf: shrinking the petacube. In: Proceeding of the 2002 ACM-SIGMOD international conference on management of data (SIGMOD'02), Madison, WI, pp 464–475

Srikant R, Agrawal R (1995) Mining generalized association rules. In: Proceeding of the 1995 international conference on very large data bases (VLDB'95), Zurich, Switzerland, pp 407–419

Srikant R, Agrawal R (1996) Mining sequential patterns: generalizations and performance improvements. In: Proceeding of the 5th international conference on extending database technology (EDBT'96), Avignon, France, pp 3–17

Srivastava J, Cooley R, Deshpande M, Tan PN (2000) Web usage mining: discovery and applications of usage patterns from web data. SIGKDD Explor 1:12–23

Steinbach M, Tan P, Kumar V (2004) Support envelopes: A technique for exploring the structure of association patterns. In: Proceeding of the 2004 ACM SIGKDD international conference on knowledge discovery in databases (KDD'04), Seattle, WA, pp 296–305

Tan P-N, Kumar V, Srivastava J (2002) Selecting the right interestingness measure for association patterns. In: Proceeding of the 2002 ACM SIGKDD international conference on knowledge discovery in databases (KDD'02), Edmonton, Canada, pp 32–41

Ting R, Bailey J (2006) Mining minimal contrast subgraph patterns. In: Proceeding of the 2006 SIAM international conference on data mining (SDM'06), Bethesda, MD, pp 638–642

Toivonen H (1996) Sampling large databases for association rules. In: Proceeding of the 1996 international conference on very large data bases (VLDB'96), Bombay, India, pp 134–145

Ukkonen A, Fortelius M, Mannila H (2005) Finding partial orders from unordered 0-1 data. In: Proceeding of the 2005 international conference on knowledge discovery and data mining (KDD'05), Chicago, IL, pp 285–293

Vanetik N, Gudes E, Shimony SE (2002) Computing frequent graph patterns from semistructured data. In: Proceeding of the 2002 international conference on data mining (ICDM'02), Maebashi, Japan, pp 458–465

Wang J, Han J (2004) BIDE: Efficient mining of frequent closed sequences. In: Proceeding of the 2004 international conference on data engineering (ICDE'04), Boston, MA, pp 79–90

Wang J, Han J, Lu Y, Tzvetkov P (2005) TFP: An efficient algorithm for mining top-k frequent closed itemsets. IEEE Trans Knowl Data Eng 17:652–664

Wang J, Han J, Pei J (2003) CLOSET+: searching for the best strategies for mining frequent closed itemsets. In: Proceeding of the 2003 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 236–245

Wang K, Jiang Y, Lakshmanan L (2003) Mining unexpected rules by pushing user dynamics. In: Proceeding of the 2003 ACM SIGKDD international conference on knowledge discovery in databases (KDD'03), Washington, DC, pp 246–255

Wang J, Karypis G (2005) HARMONY: efficiently mining the best rules for classification. In: Proceeding of the 2005 SIAM conference on data mining (SDM'05), Newport Beach, CA, pp 205–216

Wang W, Lu H, Feng J, Yu JX (2002) Condensed cube: an effective approach to reducing data cube size. In: Proceeding of the 2002 international conference on data engineering (ICDE'02), San Fransisco, CA, pp 155–165

Wang C, Wang W, Pei J, Zhu Y, Shi B (2004) Scalable mining of large disk-base graph databases. In: Proceeding of the 2004 ACM SIGKDD international conference on knowledge discovery in databases (KDD'04), Seattle, WA, pp 316–325

Wang H, Wang W, Yang J, Yu PS (2002) Clustering by pattern similarity in large data sets. In: Proceeding of the 2002 ACM-SIGMOD international conference on management of data (SIGMOD'02), Madison, WI, pp 418–427

Washio T, Motoda H (2003) State of the art of graph-based data mining. SIGKDD Explor 5:59–68

Xin D, Han J, Li X, Wah WB (2003) Star-cubing: computing iceberg cubes by top-down and bottom-up integration. In: Proceeding of the 2003 international conference on very large data bases (VLDB'03), Berlin, Germany, pp 476–487

Xin D, Han J, Shao Z, Liu H (2006) C-cubing: efficient computation of closed cubes by aggregation-based checking. In: Proceeding of the 2006 international conference on data engineering (ICDE'06), Atlanta, Georgia, p 4

Xin D, Han J, Yan X, Cheng H (2005) Mining compressed frequent-pattern sets. In: Proceeding of the 2005 international conference on very large data bases (VLDB'05), Trondheim, Norway, pp 709–720

Xin D, Shen X, Mei Q, Han J (2006) Discovering interesting patterns through user's interactive feedback. In: Proceeding of the 2006 ACM SIGKDD international conference on knowledge discovery in databases (KDD'06), Philadelphia, PA, pp 773–778

Xiong H, Shekhar S, Huang Y, Kumar V, Ma X, Yoo JS (2004) A framework for discovering co-location patterns in data sets with extended spatial objects. In: Proceeding of the 2004 SIAM international conference on data mining (SDM'04), Lake Buena Vista, FL, pp 78–89

Yan X, Cheng H, Han J, Xin D (2005) Summarizing itemset patterns: a profile-based approach. In: Proceeding of the 2005 ACM SIGKDD international conference on knowledge discovery in databases (KDD'05), Chicago, IL, pp 314–323

Yan X, Han J (2002) gSpan: graph-based substructure pattern mining. In: Proceeding of the 2002 international conference on data mining (ICDM'02), Maebashi, Japan, pp 721–724

Yan X, Han J (2003) CloseGraph: mining closed frequent graph patterns. In: Proceeding of the 2003 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 286–295

Yan X, Han J, Afshar R (2003) CloSpan: mining closed sequential patterns in large datasets. In: Proceeding of the 2003 SIAM international conference on data mining (SDM'03), San Fransisco, CA, pp 166–177

Yan X, Yu PS, Han J (2004) Graph indexing: a frequent structure-based approach. In: Proceeding of the 2004 ACM-SIGMOD international conference on management of data (SIGMOD'04), Paris, France, pp 335–346

Yan X, Yu PS, Han J (2005) Substructure similarity search in graph databases. In: Proceeding of the 2005 ACM-SIGMOD international conference on management of data (SIGMOD'05), Baltimore, MD, pp 766–777

Yan X, Zhou XJ, Han J (2005) Mining closed relational graphs with connectivity constraints. In: Proceeding of the 2005 ACM SIGKDD international conference on knowledge discovery in databases (KDD'05), Chicago, IL, pp 324–333
Yan X, Zhu F, Han J, Yu PS (2006) Searching substructures with superimposed distance. In: Proceeding of the 2006 international conference on data engineering (ICDE'06), Atlanta, Georgia, p 88
Yang G (2004) The complexity of mining maximal frequent itemsets and maximal frequent patterns. In: Proceeding of the 2004 ACM SIGKDD international conference on kowledge discovery in databases (KDD'04), Seattle, WA, pp 344–353
Yang C, Fayyad U, Bradley PS (2001) Efficient discovery of error-tolerant frequent itemsets in high dimensions. In: Proceeding of the 2001 ACM SIGKDD international conference on knowledge discovery in databases (KDD'01), San Fransisco, CA, pp 194–203
Yang LH, Lee M-L, Hsu W (2003) Efficient mining of xml query patterns for caching. In: VLDB, pp 69–80
Yang J, Wang W (2003) CLUSEQ: efficient and effective sequence clustering. In: Proceeding of the 2003 international conference on data engineering (ICDE'03), Bangalore, India, pp 101–112
Yang J, Wang W, Yu PS (2003) Mining asynchronous periodic patterns in time series data. IEEE Trans Knowl Data Eng 15:613–628
Yin X, Han J (2003) CPAR: classification based on predictive association rules. In: Proceeding of the 2003 SIAM international conference on data mining (SDM'03), San Fransisco, CA, pp 331–335
Yoda K, Fukuda T, Morimoto Y, Morishita S, Tokuyama T (1997) Computing optimized rectilinear regions for association rules. In: Proceeding of the 1997 international conference on knowledge discovery and data mining (KDD'97), Newport Beach, CA, pp 96–103
Yu JX, Chong Z, Lu H, Zhou A (2004) False positive or false negative: mining frequent itemsets from high speed transactional data streams. In: Proceeding of the 2004 international conference on very large data bases (VLDB'04), Toronto, Canada, pp 204–215
Yun U, Leggett J (2005) Wfim: weighted frequent itemset mining with a weight range and a minimum weight. In: Proceeding of the 2005 SIAM international conference on data mining (SDM'05), Newport Beach, CA, pp 636–640
Zaïane OR, Han J, Zhu H (2000) Mining recurrent items in multimedia with progressive resolution refinement. In: Proceeding of the 2000 international conference on data engineering (ICDE'00), San Diego, CA, pp 461–470
Zaki MJ (1998) Efficient enumeration of frequent sequences. In: Proceeding of the 7th international conference on information and knowledge management (CIKM'98), Washington, DC, pp 68–75
Zaki MJ (2000) Scalable algorithms for association mining. IEEE Trans Knowl Data Eng 12:372–390
Zaki M (2001) SPADE: an efficient algorithm for mining frequent sequences. Mach Learn 40:31–60
Zaki MJ (2002) Efficiently mining frequent trees in a forest. In: Proceeding of the 2002 ACM SIGKDD international conference on knowledge discovery in databases (KDD'02), Edmonton, Canada, pp 71–80
Zaki MJ, Hsiao CJ (2002) CHARM: an efficient algorithm for closed itemset mining. In: Proceeding of the 2002 SIAM international conference on data mining (SDM'02), Arlington, VA, pp 457–473
Zaki MJ, Lesh N, Ogihara M (1998) PLANMINE: sequence mining for plan failures. In: Proceeding of the 1998 international conference on knowledge discovery and data mining (KDD'98), New York, NY, pp 369–373
Zaki MJ, Parthasarathy S, Ogihara M, Li W (1997) Parallel algorithm for discovery of association rules. data mining knowl discov, 1:343–374
Zhang X, Mamoulis N, Cheung DW, Shou Y (2004) Fast mining of spatial collocations. In: Proceeding of the 2004 ACM SIGKDD international conference on knowledge discovery in databases (KDD'04), Seattle, WA, pp 384–393
Zhang H, Padmanabhan B, Tuzhilin A (2004) On the discovery of significant statistical quantitative rules. In: Proceeding of the 2004 international conference on knowledge discovery and data mining (KDD'04), Seattle, WA, pp 374–383
Zhu F, Yan X, Han J, Yu PS, Cheng H (2007) Mining colossal frequent patterns by core pattern fusion. In: Proceeding of the 2007 international conference on data engineering (ICDE'07), Istanbul, Turkey