A Comprehensive Survey of Pattern Mining: Challenges and Opportunities

Pragati Upadhyay
Research Scholar
Uttarakhand Technical University
Dehradun (Uttarakhand) India

M. K. Pandey, PhD
Director, AIMCA
Amrapali group of Institutions
Haldwani (Uttarakhand) India

Narendra Kohli, PhD
Professor & Head
Department Of CSE
Harcourt Butler Technological University, Kanpur (U.P.) India

ABSTRACT

Pattern mining is an important field of data mining. The fundamental task of data mining is to explore the database to find out sequential, frequent patterns. In recent years, data mining has shifted its focus to design methods for discovering patterns with user expectations. In this regard various types of pattern mining methods have been proposed. Frequent pattern mining, sequential pattern mining, temporal pattern mining, and constraint based pattern mining. Pattern mining has various useful real-life applications such as market basket analysis, e-learning, social network analysis, web page, click sequences, Bioinformatics, etc., this paper presents a survey of various types of pattern mining. The main goal of this paper is to present both an introduction to all pattern mining and a survey of various algorithms, challenges and research opportunities. This paper not only discusses the problems of pattern mining and its related applications, but also the extensions and possible future improvements in this field.

General Terms

Pattern mining, FP-Growth method, Frequent patterns, Sequential patterns, temporal pattern mining, D^2PM framework, TD^2PM framework

Keywords

Constraints, Sequential Pattern Mining, Frequent Pattern, Domain Driven Pattern Mining.

1. INTRODUCTION

There are various tasks of data mining such as clustering, classification, prediction, outlier analysis and pattern mining. But the major task of data mining is to discover the useful, unexpected and user expected patterns, which help to forecast the future or to understand the past [1]. However, there are several methods which discover patterns in large data that can be understood by human beings. These methods can be classified on the basis of the categories of patterns they discover. Some popular types of patterns found in the datasets are frequent item sets, different types of trends, sequential patterns, clusters and outliers and temporal patterns [2]. Agrawal and Srikant [3] has emerged this field in the 1990s with the introduction of frequent patterns. For example, patterns like {soap, cream, paste} in a transactional database of a store can be discovered by Apriori. This pattern indicates that these items are frequently purchased together.

Although several pattern mining techniques such as frequent itemset mining [3], association rule mining [3,4] are very useful for their applications, they are lacking of exploring the sequences useful in terms of time or ordering from the database. For example, to analyze the order of words in sentences, network intrusion detection, etc.. To recover this problem, sequential pattern mining was introduced. Sequential pattern mining is capable to discover in a sequential database. Sequential pattern mining is very active research filed due to its popular applications such as e-learning, web page click-stream analysis, fraud detection in digital transactions etc. Every year sequential pattern mining needs extensions. In this regard temporal pattern mining was introduced. Temporal pattern mining is a quickly developing zone of research that covers a few controls, including time arrangement investigation, pattern recognition, visualization, temporal databases, Parallel computing. Temporal data mining is connected with information mining of extensive successive informational collections. Temporal data is organized by some time point. For instance, time arrangement constitutes a well known class of consecutive information, where records are ordered by time. Some more cases of consecutive information could be arrangements of proteins, moves in a chess amusement and so forth. In this regard, temporal sequential pattern mining has been challenging field of temporal data mining. Basically, it deals with the problem of finding the frequent item sets in a given database [3]. However, it is found that without constraints, the conventional mining approaches generates a large number of patterns and rules but only few of them are useful. This phenomenon leads the concepts of constraint based pattern mining [5]. Constraints limit the number of mined patterns to reduce the complexity [4]. However, various papers regarding the survey of frequent pattern mining, sequential pattern mining and temporal pattern mining have been published, they are not presenting the relationship among these various patterns and the most recent advancements in this field. In this paper, all domains of patterns mining, challenges related to every dimension and recent techniques have been surveyed. The first section describes the frequent pattern mining, data representation used in frequent pattern mining and other type of frequent patterns. This section also provides an overview of various algorithms of frequent pattern mining, their shortcomings and comparative study of algorithms. Section three surveys about the algorithm of sequential pattern mining. In this section, most recent advancements in this domain have also been discussed. In section four constraints based sequential pattern mining methods have been discussed. Further, section five

discusses a survey of temporal pattern mining. Section six discusses the challenges and opportunities available in the field of pattern mining, and the last section conclusion is presented.

2. FREQUENT PATTERN MINING

The discovery of frequent itemsets in databases was first introduced by Agrawal and Srikant in 1993 [6]. It was initially called as mining of item sets. But in the current scenario, it is termed as frequent item set mining. Frequent item set mining was mainly proposed to understand the behavior of the customers, but it is applicable in several domains. Frequent itemset mining is useful in many applications with varied range of domains like image classification [7], network traffic analysis [8], Bioinformatics [9], detection of malware [10], analyzing the typical behavior of customers [11] and e-learning [12]. Frequent item set mining is also helpful to discover, connected patterns [13,14], designs in sequences [15,16] and graphs [17], uncommon patterns [18].

Frequent pattern mining (FPM) is initially proposed to generate interesting itemsets in the transactional database, that is having interesting relations among items. Various measures parameters can be used to analyze the interestingness of these patterns. In FPM, support is initially used to measure the interestingness of a given itemset.

Formally, the problem of FPM is defined as : Let, there be a transactional database $DB = \{T1, T2, \ldots..Tn\}$,it is a collection of transactions and I be the set of items $(I = \{i1, i2, i3, \ldots..im\})$. An itemset A is a set of items such that A ⊆ I. Let |A| denotes the number of items contained in an itemset A and |DB| denotes the cardinality of the set DB i.e. the number of transactions in a database. An itemset A is said to be length l. |A| = l, if it has l items. The support of an itemset A in a transactional database is denoted as sup (A) and defined as the number of transactions which contain this itemset A, i.e. sup $A = |\{T \mid A \subseteq T \wedge T \in DB\}|$ .Some researchers defined the support as the ratio of an itemset. That can be defined as relative $A = \sup(A) / |DB|$.

Frequent pattern mining is difficult. The beginners used to solve this problem by considering all possible itemsets that satisfy the minimum support threshold specified by the users. However, beginners' approach is not satisfactory as if the number of possible itemsets in the search space is too large or too long, this is unmanageable. To extract frequent itemsets efficiently, some more efficient methods are required that explore the search space and without considering all possible itemsets and mine them efficiently. In this regard, various algorithms have been proposed. Some of the mostly used are Apriori [1], FP-Growth method [19], Eclat [20], H-mine [21], LCM [22]. These algorithms discover the same type of frequent patterns for the same input of data, but their approaches and data structures used are different. FPM algorithms can be differentiated in:

- Which search method, they use, during discovery of patterns (BFS / DFS).
- What type of data representation they use (internal/external).
- What parameters they use to determine the itemset next to be discovered.
- What method, they apply to count the support for satisfaction of the minimum support constraint.

The next section presents the strategies used by these algorithms and discusses their advantages and disadvantages.

2.1 Searching methods (BFS and DFS)

FPM algorithms can be differentiated by the searching methods they apply to discover patterns. Most of them use either breadth first search or depth first search algorithms. BFS is called level wise algorithm. Apriori method discovers patterns using BFS. It explores database by first determining 1-l itemsets, then 2-l itemsets and so on, until it discovers the set containing all items. Whereas other algorithms such as FP-Growth, works on depth first search. DFS starts from 1-itemsets and then recursively adds items to the recent itemset list to generate large frequent itemsets.

To mine the frequent patterns efficiently it is necessary to reduce the search space. Further search space pruning methods have been proposed. For that monotone measure is used. Monotone measure can be defined as if an itmeset is not frequent, all items in its supersets are also not frequent and thus ignore them to explore, that is for any itemset A and B such that A ⊆ B, it follows that $\sup(A) \geq \sup(B)$ [3]. This property is also called as anti-monotonicity, downward-closure property or Apriori property [3]. These properties are very useful to reduce the search space.

2.2 Data representation (Horizontal and Vertical representation)

There are two standard data representation: vertical and horizontal data representation. Apriori uses a horizontal database representation. Although, Apriori is a motivation for other algorithms, it has some issues. First one is its generation of candidates by appending itemsets without focusing on database and may generate many useless patterns. Thus, it is very time consuming algorithm. Another limitation is that while using BFS, it requires more memory and in worst case complexity is O (a^2n), where a is the number of unique items and n is the count of transactions.

Eclat [23] algorithm is an improvement over Apriori. It uses a DFS search method and avoids to keep many itemsets simultaneously in memory. Eclat uses a vertical database representation. Vertical database is generated by scanning the horizontal representation of database only once. Vertical representation has very interesting properties. First is that, horizontal representation can again be regenerated through vertical representation, and the other one is that, by avoiding scanning of the database, candidates can be generated directly as well as support count can also be done directly. Eclat algorithm is a very powerful algorithm as it uses vertical

representation and possibly generates candidates and support count directly without scanning the database.

However, Eclat also has several issues. It does not scan the database and generates frequent candidates and assume any itemsets that are not present in the database, and as it uses the database where each item is represented by a unique id that is tid and it intersects the tids during candidate generation. The size of this tid list is very large that affects the runtime and memory size of Eclat. To store tid list, more memory is needed. Further Zaki and Gouda [24] introduced a new structure, diffset for vertical data representation using dEclat algorithm. This algorithm uses the difference of tidsets which has reduced the size of the itemset. It also increases the performance of operations, which are applicable on itemsets. dEclat achieves the improvement over Ecalt in memory usage and in performance specifically on dense database. However, using diffset in sparse database, is not advantageous. Further, other improvements are proposed for this tidlist as encoded bit vector tidlist [25].

2.3 Pattern Growth Algorithm

To overcome the limitations of Apriori based algorithms, pattern growth algorithms were introduced. FP- growth method, H-mine algorithm and LCM (Linear time Closed itemset Miner) algorithms are the main developments in this domain. The key idea behind pattern growth algorithm is to scan the database and find the items without candidate generation. The search is mainly based on a limited portion of the database this reduced the search space. Early pattern algorithms used projected database with DFS search. The key idea of pattern growth algorithms is to explore the frequent itemsets without including the infrequent itemsets in search space using the projected database. In this regard FREESPAN [26] algorithm was introduced and WAPMINE [27] algorithm uses the pattern method and creates WAP (Web Access Pattern) tree using the concept of tree structure mining. This method builds the WAP tree only by scanning the database twice and maintain a table called header table of frequent items with their related support. This table is used to point at the first item in frequent itemsets for each item and thus further helpful in finding the frequent sequences in a threaded way using the suffix. Although WAP mine [27] is better, but it has a major problem. As it rebuilds exclusively intermediate trees during the mining process this increases the size of resultant frequent patterns. It required more memory to store these WAP trees. Further, the PLWPP algorithm by building the trees based on prefix, not on suffix. The Prefixes are coded nodes for each position.

However, the concept of projected database increase the cost while creating multiple copies of the same database. Further, the pseudo- projection method solved this problem. It optimizes the size of the projected database by using the concept of pointers. LCM [28] algorithm merges the duplicate transactions in the projected database by using an array of support counting, hence reduces the size of the projected database. Further, H-mine [29] method used a hyper tree-structure to reduce the size of projected database as well as memory usage. FP-growth algorithm further discovered a prefix tree structure for generating the frequent itemsets. In recent years, various methods have been proposed as an improvement over the available methods to increase the efficiency of these algorithms. These algorithms have been in terms of introducing some optimized methods and enabling the frequent pattern mining algorithms to run on multi-core processors [30] and in the environment of cloud computing using Hadoop and Spark tool [31].

2.4 Other type of patterns

Although frequent pattern mining has many applications in use, it has major limitations. This section reviews some other important type of patterns which resolves the limitations of frequent pattern mining.

One major drawback of frequent mining is that the algorithm may discover a large number of itemsets and it may be difficult to identify patterns. Sometimes these itemsets may be redundant. So to represent a more concise view of frequent itemsets and reduce the amount of frequent items. Some most popular representations of frequent items have been proposed. They are the following:

(1) **Closed itemsets**- Closed itemsets [28,32] filters, frequent itemsets having no superset with the same support. It discovers only closed itemsets. The most important property of closed itemsets is its lossless representation. It represents frequent items losslesssly. It means there is no need to scan the database while recovering the information related to all frequent itemsets.

(2) **Maximal itemsets** - Maximal itemsets [33] are the set of frequent items having no frequent supersets. It means they are considered as the highest amount of frequent itemsets. Thus, they are the subset of closed itemsets. But maximal itemsets represent frequent itemsets losslessly and cannot do recovery of support of frequent item sets.

(3) Generator itemsets- Generator itemsets [34,35] is the collection of frequent sets having no subsets with the same support. This set is always equivalent or greater in size as compared to sets of closed itemsets and maximal itemsets.

3. SEQUENTIAL PATTERN MINING

Sequential Pattern mining is an active research area. Sequential pattern mining is related to the sequences. The Sequence is nothing but an ordered list of events, symbols, nominal data, etc. for example c, d, c, d, a, a, c, d represents a sequence of letters. There are various applications of sequential mining such as searching sequences of products sold out in retail stores, sequences of letters in text or speech, etc. Agrawal and Srikant [36] first introduced the concept of sequential pattern mining. Generally sequential pattern mining is related to the time series data and sequences. But in this paper, the main focus is on sequential data. Some definitions related to the sequences are discussed next here.

Let there be a set of products $I = \{I1, I2, I3, \ldots . In\}$. An itemset A is a set of products such that A ⊆ I. |A| represents the set cardinality. |A| =l denotes the length l of the itemset i.e. the number of items contained in this itemset. For

example, in a retail store $I = \{a, b, c, d, e, f\}$ represents the items purchased by the customers. The set {c, d, e,} is an itemset that contains 4 items. Thus a sequence $< \{a, b\}, \{b\}, \{c, d\}, \{e, f\} >$ represents5 transactions of customers in a retail store and represents item a, b and b are purchased at the same time, then item b, then e and f are purchased at the same time. $Sq = < .S1, S2, .S3 \ldots Sn >$ is said to be of length l, if it is having l items, for example, the sequence $< \{a, b\}, \{b\}, \{c\, d\}, \{e\, f\}, \{f\} >$ is a sequence. A sequence database is a database or a list of such sequences $SDB = < Sq1, Sq2, Sq3 \ldots\ldots Sqn >$ .SID is the sequence identifier. The Table 1 represents a sequence database that contains the sequences. Each sequence represents the transaction made by a customer.

Table 1. A Sequence Datbase

SID	Sequence
1	<{b, c},{c}, {c}, {f}>
2	<{a, b}, {c, d}, {e} >
3	<{c}, {d, e}, {e, f}>
4	<{b}, {e, f} >
5	<{e}, {d, e} >

A sequence is called a subsequence, if it is contained in another sequence. For example $< \{d\}, \{e, f\} >$ is a subsequence of $< \{c\}, \{d, e\}, \{e, f\} >$ as it is contained in that sequence. Thus the main purpose of sequential pattern mining is to explore interesting subsequences in sequential database. Originally the problem of sequential pattern mining is to find all frequent subsequence based on sequence database [30]. However, this approach is not appropriate as the number of subsequences may be very huge. So it is unrealistic and it is required to design some appropriate method to reduce the number of possible subsequences. In this regard, a number of algorithms have been proposed. These algorithms can be discussed through the methods, they use while discovering the patterns and various data structures used for the search. The next section discusses about the advantages and disadvantages of these algorithms.

Generally, these algorithms can be categorized on the basis of search techniques they use. There are two types of searching techniques to scan the database: breadth first search and depth first search. GSP algorithm is based on the BFS. On the other hand Spade [37], PrefixSpan [38], SPAM [39], Lapin [40], CM-Spam [41] and CM-Spade [41] use the DFS approach.

First, this paper discusses about GSP method. GSP scans the database level wise by first scanning the database to find 1- l sequences, then generates 2- l sequences, then 3- l sequences and so on till the last sequence. It uses a horizontal database and performs level-wise search to explore frequent patterns. It uses the Apriori property for finding frequent items. It is one of the very first sequential mining algorithms. However, it has some major limitations. The main problem with GSP is multiple database scans for calculating the support of candidates. So it is very costly. However, some optimizations can be used to reduce the cost. Another problem with GSP is that the search space is very large as it contains greater than 2^n sequences in the worst case, if a database has n number of items. Apart from this, it may discover some nonexistent patterns and it may wastage of time while considering such non existential patterns. Another big issue with GSP is its use of memory for storing all frequent patterns of length l to generate next length l+1 patterns as it uses BFS. Hence a huge consumption of memory can affect the performance. So, this method is not so appropriate.

SPADE [37] is an improvement over GSP as it uses vertical database representation. However, to represent a vertical database, it is required to scan the horizontal database once. , and horizontal database can be created by applying the reverse scan on the vertical database. SPADE takes advantage of vertical database representation. Using vertical database representation, support can be directly calculated from the ID list of patterns and ID list can be obtained without performing any scan to the original database, only by joining the ID list. SPADE algorithm can explore the whole database by scanning the database only once to prepare the ID lists of single items, then by joining the ID list, support of the pattern can be calculated. Thus, SPADE algorithm can generate all frequent patterns without scanning the database repeatedly and also there is no requirement to store a large number of patterns in memory. As a result, this algorithm is considered as the most efficient sequential pattern mining approach. However, the structure of ID list may be very large and join operation can be costly. In this regard, SPAM [38] algorithm introduced the concept of bit vectors to represent this list. Bit vector representation can reduce the requirement of memory for sequential patterns, and it is useful in dense database. Further, some techniques to compress bit vectors have been introduced. BitSpade [42] algorithm introduced bit vector approach. It is the improved version of SPADE algorithm. Another approach of indexed sparse list IDs is introduced by Fast [19] method. It is helpful in calculating the support of candidates more quickly and this reduces the amount of memory usage. Moreover, Prism [40] approach introduced the method of Prime Block Encoding. However, SPAM and BitSpade generate a large amount of candidate patterns and thus join operation is costly. Further, SPAM algorithm introduced its improved version: CM-SPAM and CM-SPADE approach [43]. They proposed the method of co-occurrence pruning [38] so that the number of joins can be reduced. For that, the database is scanned and the co-occurrence map of IDs is created. Thus, CM-SPADE is considered as the fastest algorithm [40] of sequential pattern mining. Further, Pattern Growth introduced the concept of scanning the database recursively to generate large patterns. This algorithm is DFS algorithm that avoids the problem of generating of candidate patterns that may not exist in the database. However, scanning the database recursively can be costly. Further, Pattern Growth algorithm introduced an improved version with the

concept projected database [38,40,44] which reduces the size of the search space by using DFS. PrefixSpan [38] algorithm is the most popular pattern growth algorithm, which is inspired by the FP Growth method [19]. It starts exploration of a sequential pattern consists of a single item and then recursively generate larger patterns by appending the items. It uses lexicographical order to ensure that no duplicate pattern is generated. However, the major issue of PrefixSpan is that it is very costly to scan the database repeatedly and requires more memory to store the database projections at runtime. Further, using the concept of pseudo-projection, this cost can be reduced. Pseudo-projection maintains the projected database in terms of set of pointers [45,29]. Another method of pattern growth is FreeSpan [46]. This is the previous version of PrefixSpan.

In the above sections, three types of sequential pattern mining algorithms have been discussed: BFS based algorithms that generate candidates, e.g. AprioriAll algorithm and GSP method, DFS based algorithms that generate candidates using the IDList.

4. CONSTRAINT BASED PATTERN MINING

However, sequential pattern mining has various important applications, it suffers from some fundamental limitations. The major problem with it is that the algorithm may generate a huge number of patterns depending upon the value of minimum support threshold. Moreover, as the number of pattern increases, the performance slows down in terms of runtime and memory usage. To overcome this problem, concise representation of sequential patterns has been extensively proposed. Concise representation uses only meaningful sequential patterns [47,48].Concise representation generates three types of sequential patterns: closed sequential patterns [49], maximal sequential patterns [50] and Generator sequential patterns [51]. However, these algorithms can generate more appropriate patterns, researchers proposed some constraints to reduce the number of patterns and generate more interesting patterns [52], A constraint is nothing but a special type of criteria that user impose to find more precise patterns. Various kinds of patterns have proposed. These constraints can be applied at two levels. The first way is to impose constraints after generating sequential patterns and then filter unwanted patterns. But the problem with this way is that it requires more memory and more time to generate a huge number of patterns and to store them. Another way is to impose constraint deep in the mining process. There are various constraints have been proposed.

GSP is the first algorithm to impose Gap constraint and duration constraint. Gap constraint requires that the timestamp difference between every two consecutive sequences must be either longer or shorter than the gap given in the constraint. PREFIXSpan has been extended with the integration of gap constraint by Hirate and Yamana [53]. Further item constraints have been introduced by Pei et al., with the intention to find out the presence or absence of the item in the dataset. Length constraint imposes the condition on the minimum or maximum occurrences of items in an individual sequential pattern. Whereas, aggregate constraints filter out the prices of an item in every sequence. Various aggregate constraints are available such as Sum, maximum, minimum and standard deviation etc. [54]. Another important constraint is a regular expression constraint. It allows the user to find out regular expressions present in patterns. SPIRIT algorithm uses this constraint to specify regular expressions. Three important types of constraints have been proposed that can be pushed during the mining process. Antimonotone constraint can be used to filter out the search space using the downward closure property. Second is convertible constraint. They neither belong to monotone nor antimonotone but using some techniques, they can be changed into antimonotone constraint [55]. Third is succinct constraint, it can filter out the patterns only by viewing the single item.

5. TEMPORAL PATTERN MINING

Moreover, sequential pattern mining is extended in the direction of temporal pattern mining. Various temporal pattern mining methods have been introduced to find temporal patterns in the transaction database. They partition the database on the conditions of time granularity. In this regard periodic pattern mining was introduced. It finds out the patterns that appear periodically and frequently in the database. PPM [56] and Twain [57] proposed this periodic pattern. Further calendar algebraic expression has been introduced by Ramaswamy et al. [58].

But this approach has a problem that the user should have the prior knowledge about the temporal patterns. To recover this limitation, several other approaches were proposed by the authors. But they are not able to express time intervals and periods so effectively. Further Claudia Autunes [59] proposed some additional types of constraints. ʊ-Constraints are constraints that integrate content, existential and temporal constraints. Content constraints concern with items relevant in business taxonomy. Existential constraints are related to the usual frequency parameters such as support counts. However, the algorithms produced a large number of patterns, most of them are useless and uninteresting for the end user.

To generate the patterns on user expectations, some authors have proposed to use some constraint relaxations to maintain the efficiency of the algorithm. By weakening the conditions of the original constraints, this mechanism allows the discovery of unknown information. Corresponding to this field, Domain Driven Pattern Mining (D^2PM) framework was proposed [60] that allows for the discovery of patterns flexible in regards of transactional, sequential and structured patterns. These patterns are derived under constraints resultant from ontologies that capture background knowledge. This framework uses domain knowledge, represented through domain ontology. This framework adapted the Interleaved algorithm to deal with the constraints defined, namely timespan constraint, complete constraint and cyclic constraints. D^2PM framework distinguishes between two types of temporal constraints based on the time ontology. First is timespan constraints and other is cyclic temporal constraints. Cyclic constraints can be categorized into

complete cyclic temporal constraints and partial cyclic temporal constraints.

Further, this framework is extended to incorporate time constraints deep into mining process and deal with complete periodicity. TD^{2}PM standing for Temporality in Transactional Domain-Driven Pattern Mining [61] is the result of this extension which includes timespan and partial cyclic constraints. This algorithm allows for the discovery of patterns according to any time granularity chosen, without any further pre-processing. Further Walaa N. Ismail [21] developed an algorithm named PPFP-growth (Productive Periodic –Frequent Pattern-growth) that is able to extract all productive-associated periodic-frequent patterns. That is very helpful in generating pattern of various health-related vital sign data obtained from body sensor network in healthcare.

6. CHALLENGES AND OPPORTUNITIES

Finding out various types of patterns (frequent, sequential, temporal, etc.) is still an emerging and active research field. Although various methods have been studied and proposed, more scope is remaining. We point out some important opportunities:

Requirement of more efficient algorithm- Any type of pattern mining is expensive. Its run time and memory requirement is costly. Especially, in case of dense databases where long sequences are present. Although several methods have been proposed and recent methods are more efficient than older methods, there is requirement of improvement. Some challenging areas such as high utilization patterns, uncertain patterns, designing of parallel, multi-core, distributed, GPU based methods are still to explore [62].

Requirement of methods to deal with more complex data- Another important challenge is to handle more complex data while mining the patterns. The Major challenge in this dimension is Spatial patterns [16]. Another issue is to explore complex patterns in databases. Further to search more interesting patterns, research should be required to design interesting measures.

Applications- Since temporal sequential data is present in many new fields, there are various opportunities is in this dimension. The most promising applications in this field are social network analysis, fraud detecting in digital transactions, internet of things and sensor networks etc.

7. CONCLUSION

Finally, it can be concluded that Pattern mining is an emerging and the most challenging areas for which many stimulating problems remain open. The inclusion of constraints to existing pattern mining techniques may be used as a tool for a recommendation system or for advertising targeting purposes, analyzing web log servers [32], financial and biomedical purposes among other application. This paper has presented a comprehensive survey of various pattern mining techniques. The entire literature is viewed in three dimensions, first the type of patterns that have to be mined: sequential patterns, frequent patterns, temporal patterns, constraint based patterns, and second their shortcomings and recent advancements and the lastly various challenges and opportunities related to this field. In addition, the paper has discussed other research problems related every pattern mining such as enhancing the frequency of patterns, reducing the memory usage etc.

8. REFERENCES

[1] Aggarwal CC,"Data mining: the textbook," Heidelberg: Springer, 2015.

[2] Han J, Pei J, Kamber M, "Data mining: concepts and techniques," Amsterdam: Elsevier, 2011.

[3] Agrawal R, Srikant, "R. Fast algorithms for mining association rules," In: Proc. 20th int. conf. very large data bases, VLDB 1994, Santiago de Chile, Chile, pp.487-499, 12-15 September 1994.

[4] Antunes C, and Oliveira A, "Sequential Pattern Mining with Approximated Constraints," in Proceedings of the International Conference on Applied Computing, pp. 131-138, 2004.

[5] Hu Y, "The Research of Customer Purchase Behavior using Constraint-Based Sequential Pattern Mining Approach," Thesis Report, National Central University Library Electronic Theses & Dissertations System, 2007.

[6] Agrawal R, Srikant, "R. Fast algorithms for mining association rules," In: Proc. 20th int. conf. very large data bases, VLDB 1994, Santiago de Chile, Chile, pp.487-499, 12-15 September 1994.

[7] Fernando B, Elisa F, Tinne T, "Effective use of frequent itemset mining for image classification," In: European Conference on Computer Vision, Florence, Italy, pp. 214-227, 7-13 October,2012.

[8] Glatz E, Mavromatidis S, Ager B, Dimitropoulos X, "Visualizing big network track data using frequent pattern mining and hypergraphs,", Computing, 96(1), pp. 27-38, 2014.

[9] Duan Y, Fu X, Luo B, Wang Z, Shi J, Du X, "Detective Automatically identify and analyze malware processes in forensic scenarios via DLLs," IEEE International Conference on Communications, London, United Kingdom, pp. 5691-5696, 8-12 June, 2015.

[10] Mukherjee Liu, Glance, "Spotting fake reviewer groups in consumer reviews," In: Proc. 21st international conference on World Wide Web, Lyon, France, pp. 191-200, 16-20 April, 2012.

[11] Liu Y, Zhao Y, Chen L, Pei J, Han J, "Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays," IEEE Transactions on Parallel and Distributed Systems, 23(11), pp. 2138-2149, 2012.

[12] Mwamikazi E, Fournier-Viger P, Moghrabi C, Baudouin R, "A Dynamic Questionnaire to Further Reduce Questions in Learning Style Assessment," In: Proc. 10th Int. Conf.Artificial Intelligence Applications and Innovations, Rhodes, Greece, pp. 224-235, 19-21 September, 2014.

[13] Fournier-Viger P, Lin J C W, Dinh T, Le HB, "Mining Correlated High-Utility Itemsets using the Bond Measure.", In: Proc. Intern.Conf. Hybrid Artificial Intelligence Systems Seville, Spain, pp.53-65, 18-20 April, 2016.

[14] Soulet A, Raissi C, Plantevit M, Cremilleux B, "Mining dominant patterns in the sky," In: Proc. 11th IEEE Int. Conf. on Data Mining, Vancouver, Canada, pp. 655-664, 11-14 December,2011.

[15] Mabroukeh NR, Ezeife CI, "A taxonomy of sequential pattern mining algorithms," ACM Computing Surveys, 43(1): 3, 2010.

[16] Fournier-Viger P, Gomariz A, Campos M, Thomas R, "Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information," In: Proc. 18th Pacific-Asia Conf. Knowledge Discovery and Data Mining. Tainan, Taiwan, pp. 40-52, 13-16 May, 2014.

[17] Yan X, Han J, "gspan: Graph-based substructure pattern mining," In: Proc. 2002 Intern. Conf. Data Mining, Maebashi City, Japan, pp. 721-724, 9-12 December, 2002.

[18] Koh Y S, Ravana S R, "Unsupervised Rare Pattern Mining: A Survey," ACM Transactions on Knowledge Discovery from Data, 10(4): article no. 45, 2016.

[19] Han J, Pei J, Ying Y, Mao R, "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," Data Min. Knowl. Discov. 8(1), pp. 53-87, 2004.

[20] Zaki M J, "Scalable Algorithms for Association Mining," IEEE Trans. Knowl. Data Eng., 12(3), pp. 372-390, 2000.

[21] Pei J, Han J, Lu H, Nishio S, Tang S, Yang D, "H-mine: Hyper-structure mining of frequent patterns in large databases," In: Proc. IEEE Intern. Conf. Data Mining, San Jose, USA, pp. 441-448, 29 November - 2 December, 2001.

[22] Uno T, Kiyomi M, Arimura H, "LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets", Proc. ICDM'04 Workshop on Frequent Itemset Mining Implementations, CEUR, 2004.

[23] Zaki M J, Gouda K, "Scalable Algorithms for Association Mining," IEEE Trans. Knowl. Data Eng., 2000.

[24] Zaki M J, Gouda K, "Fast vertical mining using diffsets", In: Proc. 9th ACM SIGKDD Intern. Conf. Knowledge Discovery and Data Mining, Washington DC, USA, pp. 326-335, 24 – 27 August, 2003.

[25] Lucchese C, Orlando S, Perego R, "Fast and Memory Efficient Mining of Frequent Closed Itemsets," IEEE Trans. Knowl. Data Eng., 18(1), pp. 21-36, 2006.

[26] Han J, Dong G, Mortazavi-Asl B, Chen Q, Dayal U, Hsu M C, "Freespan: Frequent pattern-projected sequential pattern mining," Proceedings 2000 Int. Conf. Knowledge Discovery and Data Mining (KDD'00), pp. 355-359, 2000.

[27] Myra S, "Web usage mining for Web site evaluation," Communications of the ACM, vol. 43, No. 8, pp. 127–134, 2000.

[28] Uno T, Kiyomi M, and Arimura H, "LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets," Proc. ICDM'04, Workshop on Frequent Itemset Mining Implementations, CEUR, 2004.

[29] Pei J, Han J, Lu H, Nishio S, Tang S, Yang D, "H-mine: Hyper-structure mining of frequent patterns in large databases," In: Proc. IEEE Intern. Conf. Data Mining, San Jose, USA, pp. 441-448, 29 November - 2 December, 2001.

[30] Srikant R, and Agrawal R, "Mining sequential patterns: Generalizations and performance improvements," The International Conference on Extending Database Technology, pp. 1-17, 1996.

[31] Aliberti G, Colantonio A, Di Pietro R, Mariani R., "EXPEDITE: EXPress closed ITemset Enumeration," Expert Systems with Applications, 42(8), pp. 3933-3944, 2015.

[32] Suguna K, "Frequent Pattern Mining of Web Log Files Working Principles," vol. 157, no. 3, pp. 1–5, 2017.

[33] Vo B, Hong TP, Le B, "DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets," Expert Systems with Applications, 39(8), pp. 7196-206, 2012.

[34] Szathmary L, Valtchev P, Napoli A, Godin R, Boc A, Makarenkov V, "A fast compound algorithm for mining generators, closed itemsets, and computing links between equivalence classes," Annals of Mathematics and Artificial Intelligence, pp. 81-105, 2014.

[35] Fournier-Viger P, Wu CW, Tseng VS, "Novel concise representations of high utility item-sets using generator patterns," In: Proc. Intern. Conf. International Conference on Advanced Data Mining and Applications, Guilin, China, pp. 30-43, 19-21 December, 2014.

[36] Srikant R, and Agrawal R, "Mining sequential patterns: Generalizations and performance improvements," The International Conference on Extending Database Technology, pp. 1-17, 1996.

[37] Zaki M. J., "SPADE: An efficient algorithm for mining frequent sequences," Machine learning, vol.42 (1-2), pp. 31-60, 2001.

[38] Pei J, Han J, Mortazavi-Asl B, Wang J, Pinto H, Chen Q, Dayal U, and Hsu M. C., "Mining sequential patterns by pattern-growth: The prefixspan approach," IEEE Transactions on knowledge and data engineering, vol. 16(11), pp. 1424-1440, 2004.

[39] Ayres J, Flannick J, Gehrke J, and Yiu T, "Sequential pattern mining using a bitmap representation," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.429-435, 2002.

[40] Fournier-Viger P, Gomariz A, Campos M, and Thomas R, "Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information," The Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2014.

[41] Yang Z, and Kitsuregawa M, "LAPIN-SPAM: An improved algorithm for mining sequential pattern," The International Conference on Data Engineering Workshops, pp. 1222-1222, 2005.

[42] Aseervatham S, Osmani A, and Viennet E, "bitSPADE: A lattice-based sequential pattern mining algorithm using bitmap representation," The International Conference on Data Mining, pp. 792-797, 2006.

[43] Yang Z, and Kitsuregawa M, "LAPIN-SPAM: An improved algorithm for mining sequential pattern," The International Conference on Data Engineering Workshops, pp. 1222-1222, 2005.

[44] Fournier-viger P, Lin J C, "A Survey of Itemset Mining," pp. 1–41, 2017.

[45] Han J, Pei J, Ying Y, and Mao R, "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," Data Mining and Knowledge Discovery, vol. 8(1), 2004.

[46] Han J, Pei J, Mortazavi-Asl B, Chen Q, Dayal U, and Hsu M C, "FreeSpan: frequent pattern projected sequential pattern mining," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 355-359, 2000.

[47] Huang K Y, Chang C H, Tung J H, and Ho C T, "COBRA: closed sequential pattern mining using bi-phase reduction approach," The International Conference on Data Warehousing and Knowledge Discovery, pp. 280-291, 2006.

[48] Ge J, Xia Y, and Wang J, "Towards efficient sequential pattern mining in temporal uncertain databases", The Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 268-279, 2015.

[49] Wang J, Han J, and Li C, "Frequent closed sequence mining without candidate maintenance, "IEEE Transactions on Knowledge Data Engineering, vol. 19(8), pp. 1042-1056, 2007.

[50] Gomariz A, Campos M, Marin R, and Goethals B, "ClaSP: "An efficient algorithm for mining frequent closed sequences," The Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 50-61, 2013.

[51] Pham T T, Luo J, Hong T P, and Vo B, "MSGPs: a novel algorithm for mining sequential generator patterns," The International Conference on Computational Collective Intelligence, pp.393-401, 2012.

[52] Zhang J, Wang Y, and Yang D, "CCSpan: Mining closed contiguous sequential patterns,", Knowledge-Based Systems, vol. 89, pp.1-13, 2015.

[53] Yu H, Yamana H, "Generalized sequential pattern mining with item intervals," Journal of Computers, vol. 1(3), pp. 51-60, 2006.

[54] Pei J, Han J, and Wang W, "Constraint-based sequential pattern mining: the pattern-growth methods," Journal of Intelligent Information Systems, vol. 28(2), pp. 133-160, 2007.

[55] Pei J, Han J, and Lakshmanan L V, "Mining frequent itemsets with convertible constraints," The International Conference on Data Engineering, pp.433-442, 2001.

[56] Lee C, Chen M, Lin C, "Progressive partition miner: an efficient algorithm for mining general temporal association rules," IEEE Transaction on Knowledge and Data Engineering 15(4), PP. 1004–1017 (2003).

[57] Huang J, Dai B, Chen M, "Twain: Two-End Association Miner with Precise Frequent Exhibition Periods," ACM Transactions on Knowledge Discovery from Data mining, 1(2), 2007.

[58] Ramaswamy S, Mahajan S, Silberschatz A, "On the Discovery of Interesting Patterns in Association Rules," In: International Conference on Very Large Databases, New York, USA, pp. 368–379, 1998.

[59] Antunes C, "Pattern Mining over Nominal Event Sequences using Constraint Relaxations," Ph.D. Thesis, Instituto Superior Técnico, Lisboa, Portugal, January 2005.

[60] Antunes C M, "D2pm: a framework for mining generic patterns. Technical, Instituto Superior Technical, Lisbon, 2011.

[61] Pina S M, and Antunes C, "(TD) 2 PaM : A Constraint-Based Algorithm for Mining Temporal Patterns in Transactional Databases," no. i, pp. 390–407, 2013.

[62] Fournier-viger P, and J. C. Lin, "A Survey of Sequential Pattern Mining," vol. 1, no. 1, pp. 54–77, 2017.