Vandana Publications

Volume-4, Issue-1, February-2014, ISSN No.: 2250-0758

International Journal of Engineering and Management Research

Available at: www.ijemr.net

Page Number: 241-248

A Study of Sequential Pattern Mining Techniques

Kapil Sharma[1], Ashok[2], Dr. Harish Rohil[3]

[1]M.Tech (Scholar), Department of Computer Sc. and Applications, Ch. Devi Lal University, Sirsa, Haryana, INDIA

[2]M.Tech (Scholar), Department of Computer Sc. and Applications, Ch. Devi Lal University, Sirsa, Haryana, INDIA

[3]Asst. Professor, Department of Computer Sc. and Applications, Ch. Devi Lal University, Sirsa, Haryana, INDIA

ABSTRACT

A lot of database is available in the world which is to be screened to find out various facts such as history of symptoms from which disease can be diagnosed, accessing data from Knowledge Data Discovery. If we try to filter these data manually, it can take hours or days or months. So we need some interesting tool to access or retrieve the data for which data mining is used to mine that data and to access that mined data we have several techniques and among them sequential pattern mining is better. This paper presents a study of sequential pattern techniques.

Keywords: Apriori, Data Mining, FREESPAM, GSP, Sequential Pattern Mining, SPIRIT, SPADE, WSPAN, Pattern Growth

I. INTRODUCTION

1.1 Data Mining

The techniques that perform data analysis and may uncover important data patterns are needed. One of these techniques is known as data mining. Data mining refers to extracting or mining knowledge from large amounts of data [32].

1.2 Data Mining Models

The data mining models are of two types: Predictive and Descriptive. The predictive model makes prediction about unknown data values by using the known values. Ex. Classification, Regression, Time series analysis, Prediction etc. The descriptive model identifies the patterns or relationships in data and explores the properties of the data examined. Ex. Clustering, Summarization, Association rule, Sequence discovery etc.

1.3 Various Technique of Data Mining

There are several major data mining techniques that have been developed and used in data mining projects recently including association, classification, clustering, prediction and sequential patterns. They are briefly explained as:

a) Association

Association is one of the best known data mining technique. In association, a pattern is discovered based on a relationship of a particular item on other items in the same transaction. For example, the association technique is used in market basket analysis to identify what products that customers frequently purchase together. Based on this data businesses can have corresponding marketing campaign to sell more products to make more profit.

b) Classification

Classification is a classic data mining technique based on machine learning. Basically classification is used to classify each item in a set of data into one of predefined set of classes or groups. Classification method makes use of mathematical techniques such as decision trees, linear programming, neural network and statistics. In classification, make the software is made that can learn how to classify the data items into groups. For example, we can apply classification in application that given all past records of employees who left the company, predict which current employees are probably to leave in the future. In this case, we divide the employee's records into two groups that are leave and stay. And then it can be asked from the data mining software to classify the employees into each group.

C) Clustering

Clustering is a data mining technique that makes meaningful or useful cluster of objects that have similar characteristic using automatic technique. Different from classification, clustering technique also defines the classes and put objects in them, while in classification objects are assigned into predefined classes. To make the concept clearer, here is an example of library. In a library, books have a wide range of topics available. The challenge is how to keep those books in a way that readers can take several books in a specific topic without hassle. By using clustering technique, books can be kept that have some kind of similarities in one cluster or one shelf and label it with a meaningful name. If readers want to grab books in a topic, only go to that shelf instead of looking the whole in the whole library.

d) Prediction

The prediction as it name implied is one of a data mining techniques that discovers relationship between independent variables and relationship between dependent and independent variables. For instance, prediction analysis technique can be used in sale to predict profit for the future if sale is considered as an independent variable; profit could be a dependent variable. Then based on the historical sale and profit data, a fitted regression curve can be drawn that is used for profit prediction.

II. SEQUENTIAL PATTERNS

Sequential patterns analysis is one of data mining technique that seeks to discover similar patterns in data transaction over a business period. The uncover patterns are used for further business analysis to recognize relationships among data. Sequential patterns analysis in one of data mining technique that seeks to discover similar patterns in data transaction over a business period. The uncover patterns are used for further business analysis to recognize relationships among data. Sequential pattern mining is trying to find the relationships between occurrences of sequential events by looking for any specific order of occurrences. In other words, sequential pattern mining is aiming at finding the frequently occurred sequences to describe the data or predict future data or mining periodical patterns. Sequential pattern is a sequence of item sets that frequently occurred in a specific order, all items in the same item sets are supposed to have the same transaction-time value or within a time gap [20].

Sequential pattern mining can be divided into two parts according to the way by which candidate sequences are generated and stored. And the way in which support is counted and how candidate sequences are tested for frequency.

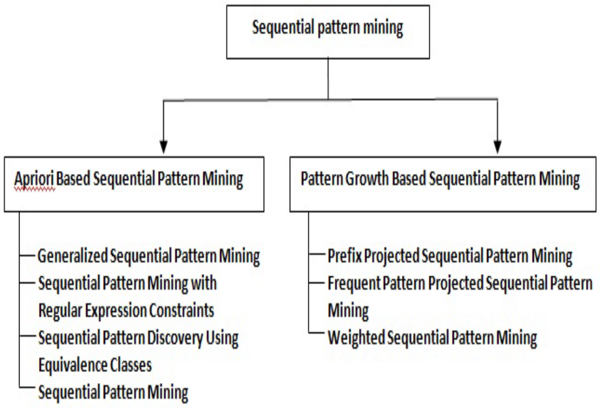

Sequential pattern mining
Apriori Based Sequential Pattern Mining
Pattern Growth Based Sequential Pattern Mining
Generalized Sequential Pattern Mining
Sequential Pattern Mining with Regular Expression Constraints
Sequential Pattern Discovery Using Equivalence Classes
Sequential Pattern Mining
Prefix Projected Sequential Pattern Mining
Frequent Pattern Projected Sequential Pattern Mining
Weighted Sequential Pattern Mining

Fig.1 Sequential Pattern Mining Taxonomy

III. APRIORI BASED SEQUENTIAL PATTERN MINING

The apriori property states that all nonempty subsets of a frequent item set must also be frequent. It is also described as ant monotonic in that if a sequence cannot pass the minimum support test, its entire super sequences will also fail the test. Characteristics of Apriori-based Method are:

Breadth-First Search: Apriori-based algorithms are described as breath-first (level-wise) search algorithms because they construct all the k-sequences, in k^{th} iteration of the algorithm, as they traverse the search space [20] [21].

Generate-and-Test: This feature is used by the very early algorithms in sequential pattern mining. Algorithms that depend on this feature only display an inefficient pruning method and generate an explosive number of candidate sequences and then test each one by one for satisfying some user specified constraints, consuming a lot of memory in the early stages of mining.

Multiple Scans of The Database: This feature entails scanning the original database to ascertain whether a long list of generated candidate sequences is frequent or not. It is a very undesirable characteristic of most apriori-based algorithms and requires a lot of processing time and I/O cost.

3.1 Generalized Sequential Pattern Mining (GSP):

GSP is not a main-memory algorithm. It is an extension of their seminal algorithm for frequent item set mining, known as Apriori [1] [21]. GSP uses the downward-closure property of sequential patterns and adopts a multiple-pass, candidate generate-and-test approach. If the candidates do not fit in memory, the algorithm generates only as many candidates as will fit in memory and the data is scanned to count the support of these candidates.

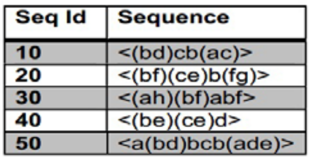
Frequent sequences resulting from these candidates are written to disk, while those candidates without minimum support are deleted. This procedure is repeated until all the candidates have been counted [8]. In the first scan of the database, it finds all of the frequent items, that is, those with minimum support.

Each such item yields a 1-event frequent sequence consisting of that item. Each subsequent pass starts with a seed set of sequential patterns—the set of sequential patterns found in the previous pass [27]. This seed set is used to generate new potentially frequent patterns, called candidate sequences. Each candidate sequence contains one more item than the seed sequential pattern from which it was generated (where each event in the pattern may contain one or multiple items). Recall that the number of instances of items in a sequence is the length of the sequence. So, all of the candidate sequences in a given pass will have the same length. We refer to a sequence with length k as a k-sequence. Let Ck denote the set of candidate k-sequences. A pass over the database finds the support for each candidate k-sequence. The candidates in Ck with at least min sup form Lk, the set of all frequent k-sequences. This set then becomes the seed set for the next pass, k + 1. The algorithm terminates when no new sequential pattern is found in a pass, or no candidate sequence can be generated.

Example of GSP: Consider the database [17] as our problem is to find all frequent sequences, given min_sup=2.

Table 1.

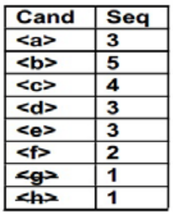
Sample Database [17]



Seq Id	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Table 2.

Length-1

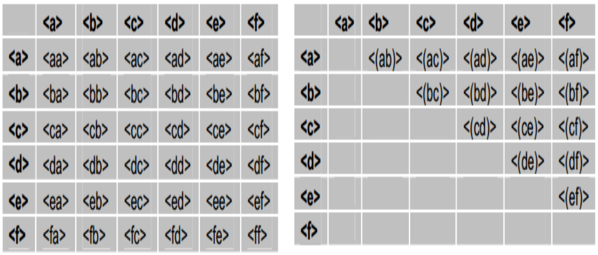


Cand	Seq
<a>	3
<b>	5
<c>	4
<d>	3
<e>	3
<f>	2
<g>	1
<h>	1

.

In table 1 there is a database having Seq ID and sequences. On the basis of min support which is 2 filter the data as in table2.

Table 3.Length-2 Candidates



	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

	<a>	<b>	<c>	<d>	<e>	<f>
<a>		<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
<b>			<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>				<(cd)>	<(ce)>	<(cf)>
<d>					<(de)>	<(df)>
<e>						<(ef)>
<f>						

As shown in Table 3, using Apriori one needs to generate just 51 length-2 candidates, while without Apriori property, 8*8+8*7/2=92 candidates would need to be generated. For this example, Apriori would perform 5 database scans, pruning away candidates with support less than min sup. Candidates that cannot pass support threshold are pruned.

1st scan: 8 candidates. 6 length-1 sequence patterns.

2nd scan: 51 candidates. 19 length-2 sequence patterns. 10 candidates not in DB at all

3rd scan: 46 candidates. 19 length-3 sequence patterns. 20 candidates not in DB at all

4th scan: 8 candidates. 6 length-4 sequence patterns.

5th scan: 1 candidate. 1 length-5 sequence patterns.

3.2 Sequential Pattern Mining With Regular Expression Constraints (SPIRIT):

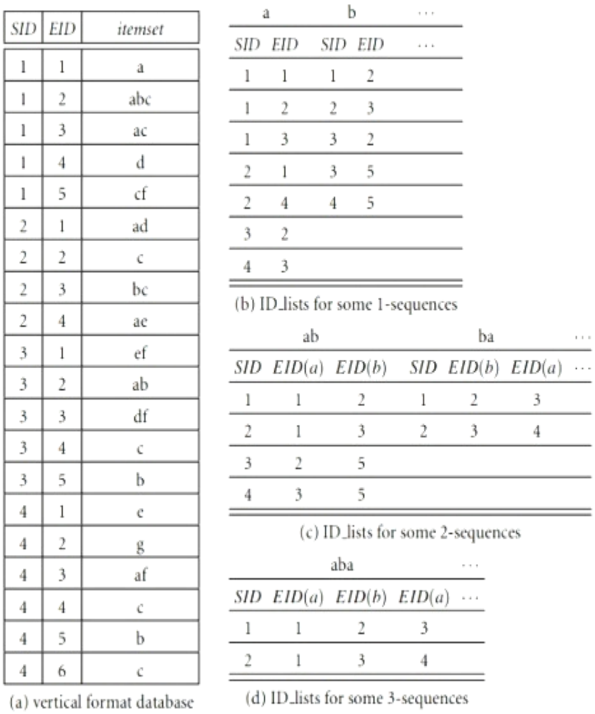
It is also known as constraint based sequential pattern mining. Constraints can be expressed in many forms [5][8][21]. They may specify desired relationships between attributes; attribute values, or aggregates within the resulting patterns mined. Regular expressions can also be used as constraints in the form of “pattern templates,” which specify the desired form of the patterns to be mined. The key idea to note is that these kinds of constraints can be used during the mining process to confine the search space, thereby improving (1) the efficiency of the mining and (2) the interestingness of the resulting patterns found. This idea is also referred to as “pushing the constraints deep into the mining process.”SPIRIT is a family of algorithms for sequential pattern mining with regular expression constraints. Its general idea is to use some relaxed constraint which has nice property to prune [3]. There exist several versions of the algorithm, differing in the degree to which the constraints are enforced to prune the search space of pattern during computation [9]. The main distinguishing factor among the schemes is the degree to which the regular expression constraints are enforced to prune the search space. There are several categories for constraint based mining:

1. **Item constraint:**
   An item constraint specifies subset of items that should or should not be present in the patterns.
2. **Length constraint:**
   A length constraint specifies the requirement on the length of the patterns, where the length can be either the number of occurrences of items or the number of transactions.
3. **Super-pattern constraint:**
   Super-patterns are ones that contain at least one of a particular set of patterns as sub-patterns.
4. **Aggregate constraint:**
   An aggregate constraint is the constraint on an aggregate of items in a pattern, where the aggregate function can be sum, avg, max, min, standard deviation, etc.
5. **Regular expression constraint:**
   A regular expression constraint CRE is a constraint specified as a regular expression over the set of items using the established set of regular expression operators, such as disjunction and Kleene closure.

3.3 Sequential Pattern Discovery using Equivalence Classes (SPADE):

It is also called vertical format based sequential pattern mining. It is an efficient sequential pattern mining algorithm based on vertical format. It utilizes combinatorial properties to decompose the original problem into smaller sub-problems [19][21]. SPADE use a vertical id list database format and use a lattice-theoretic approach to decompose the original search space (lattice) into smaller pieces (sub-lattices) which can be processed independently in main-memory. All sequences are discovered in **three database scans**, or only a single scan with some Pre-processed information. SPADE not only minimizes I/O costs by reducing database scans, but also minimizes computational costs by using efficient search schemes. In real world applications different items may have different support threshold to describe whether a given item or item set is a frequent item set [14]. This means each item will contain its own support threshold depends upon various issues like cost of item, environmental factors etc. The use of vertical data format, with the creation of ID lists, reduces scans of the sequence database. The ID lists carry the information necessary to find the support of candidates. As the length of a frequent sequence increases, the size of its ID list decreases, resulting in very fast joins.

Table 4. SPADE Example [8]



SID	EID	itemset
1	1	a
1	2	abc
1	3	ac
1	4	d
1	5	cf
2	1	ad
2	2	c
2	3	bc
2	4	ae
3	1	ef
3	2	ab
3	3	df
3	4	c
3	5	b
4	1	e
4	2	g
4	3	af
4	4	c
4	5	b
4	6	c

(a) vertical format database

a		b		···
SID	EID	SID	EID	···
1	1	1	2	
1	2	2	3	
1	3	3	2	
2	1	3	5	
2	4	4	5	
3	2			
4	3			

(b) ID_lists for some 1-sequences

ab			ba			···
SID	EID(a)	EID(b)	SID	EID(b)	EID(a)	···
1	1	2	1	2	3	
2	1	3	2	3	4	
3	2	5				
4	3	5				

(c) ID_lists for some 2-sequences

aba				···
SID	EID(a)	EID(b)	EID(a)	···
1	1	2	3	
2	1	3	4	

(d) ID_lists for some 3-sequences

As shown in table 4. let min sup = 2. Our running example sequence database, S, is in horizontal data format. SPADE first scans S and transforms it into vertical format. Each itemset (or event) is associated with its ID list, which is the set of SID (sequence ID) and EID (event ID) pairs that contain the itemset. The ID list for individual items, a, b, and so on. For example, the ID list for item b consists of the following (SID, EID) pairs: f(1, 2), (2, 3), (3, 2), (3, 5), (4, 5)g, where the entry (1, 2) means that b occurs in sequence 1, event 2, and so on. Items a and b are frequent. They can be joined to form the length-2 sequence, ha, bi. We find the support of this sequence as follows. We join the ID lists of a and b by joining on the same sequence ID wherever, according to the event IDs, a occurs before b. That is, the join must preserve the temporal order of the events involved. The result of such a join for a and b is shown in the ID list for ab. For example, the ID list for 2-sequence ab is a set of triples, (SID, EID (a), EID(b)), namely f(1, 1, 2), (2, 1, 3), (3, 2, 5), (4, 3, 5)g. The entry (2, 1, 3), for example, shows that both a and b occur in sequence 2, and that a (event 1 of the sequence) occurs before b (event 3), as required. Furthermore, the frequent 2-sequences can be joined (while considering the Apriori pruning heuristic that the (k-1)-subsequences of a candidate k-sequence must be frequent) to form 3-sequences and so on. The process terminates when no frequent sequences can be found or no candidate sequences can be formed [8].

3.4 Sequential Pattern Mining (SPAM)

It is a new algorithm for finding all frequent sequences within a transactional database. The algorithm is especially efficient when the sequential patterns in the database are very long. A depth-first search strategy is used to generate candidate sequences and various pruning mechanisms are implemented to reduce the search space. The transactional data is stored using a vertical bitmap representation, which allows for efficient support counting as well as significant bitmap compression. It incrementally outputs new frequent item sets in an online fashion. Yet how to efficiently implement the mining is difficult due to the inherent characteristic of the problem - the large size of the data set. So for getting the entire frequent pattern from a large database [21].

Summary of Apriori Based Approaches

Apriori based approaches are easy to implement and use large itemset property. All the approaches of apriori methods are described in table 5.

Table 5. Summarized Apriori Based Methods

S.No	Algorithm	Category	Key Feature	Merits/ Demerits
1	GSP	Apriori Based approach	Apriori and BFS based approach, use downward closure Property	Merits: Uses large itemset property. Easily parallelized Easy to implement Demerits: Assumes transaction database is memory resident. Requires many database scans.
2	SPIRIT		Use regular expression constraints for pruning, BFS based approach	
3	SPADE		Use lattice-theoretic based approach,	
4	SPAM		using a vertical bitmap representation for data storage.	

IV. PATTERN GROWTH BASED SEQUENTIAL PATTERN MINING

Pattern growth based divides the sequential patterns to be mined based on the subsequences obtained so far and project the sequence database based on the partition of such patterns [7]. Such a methodology is called sequential pattern mining by pattern growth. The general idea is outlined as follows: Instead of repeatedly scanning the entire database and generating and Testing large sets of candidate sequences, one can recursively project a sequence database into a set of smaller databases associated with the set of patterns mined so far and, then, mine locally frequent patterns in each projected database. Characteristics of Pattern Growth-Based Methods are:

Sampling And/or Compression: The problem with sampling is that the support threshold must be kept small, which causes a combinatorial explosion in the number of candidate patterns. We think that sampling should be given more attention in pattern-growth and early-pruning algorithms as a way to reduce search space and processing time for mining. An open research area is in investigating ways to sample sequences and partition the search space based on the Fibonacci sequence as a guide for sampling or partitioning of sequences. Another way to reduce the search space while mining sequential patterns is to focus on concise representations of sequences, such as mining maximal or closed sequences.

Candidate Sequence Pruning: Pattern-growth algorithms try to utilize a data structure that allows them to prune candidate sequences early in the mining process. This result in early display of smaller search space and maintain a more directed and narrower search procedure.

Search Space Partitioning: It allows partitioning of the generated search space of large candidate sequences for efficient memory management. There are different ways to partition the search space. Once the search space is partitioned, smaller partitions can be mined in parallel. Advanced techniques for search space partitioning include projected databases and conditional search, referred to as split-and-project techniques.

Depth-First Traversal: That depth-first search of the search space makes a big difference in performance, and also helps in the early pruning of candidate sequences as well as mining of closed sequences. The main reason for this performance is the fact that depth-first traversal utilizes far less memory, more directed search space, and thus less candidate sequence generation than breadth-first or post-order which are used by some early algorithms [9].

Tree Projection: Tree projection usually accompanies pattern-growth algorithms. Here, algorithms implement a physical tree data structure representation of the search space, which is then traversed breadth-first or depth-first in search of frequent sequences, and pruning is based on the apriori property.

Suffix/Prefix Growth: Projected databases and conditional search in trees first find the frequent 1-sequences, and hold each frequent item as prefix or suffix, then start building candidate sequences around these items and mine them recursively. The idea behind this is that frequent subsequences can always be found by growing a frequent prefix/suffix; since it is usually shared among a good number of these sequences. This greatly reduces the amount of memory required to store all the different candidate sequences that share the same prefix/suffix [4].

4.1 PREFIXSPAN:

(Prefix-projected Sequential pattern mining): Prefix Span is a new pattern-growth method for mining sequential patterns and explores prefix projection in sequential pattern mining. Its major idea is that, instead of projecting sequence databases by considering all the possible occurrences of frequent subsequences, the projection is based only on frequent prefixes because any frequent subsequence can always be found by growing a frequent prefix. The complete set of sequential patterns partitioned into the subsets according to the prefixes. The subsets of sequential patterns can be mined by constructing corresponding projected databases and mine each recursively. PrefixSpan mines the complete set of by employing the divide-and-conquer strategy and greatly reduces the efforts of candidate subsequence generation. The first scan of the database derives the set of length-1 sequential patterns [6]. 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. Its general idea is to examine only the frequent prefix subsequences and project only their corresponding postfix subsequences into projected databases because any frequent subsequence can always be found by growing a frequent prefix. No candidate sequence needs to be generated by PrefixSpan. Projected databases keep shrinking. The major cost of PrefixSpan is the construction of projected databases. To further improve mining efficiency, two kinds of database projections are explored: level-by-level projection and bi-level projection. Moreover, a main-memory-based pseudo-projection (using pointers rather than physically copying postfix sequences) technique is developed for saving the cost of projection and speeding up processing when the projected (sub)-database and its associated pseudo-projection processing structure can fit in main memory [4][21].

Additionally, PrefixSpan is efficient because it mines the complete set of patterns and has a significantly faster running than both GSP algorithm and FreeSpan. The major cost of PrefixSpan, similarly to FreeSpan, is the construction of projected databases. At worst, for every sequential database, PrefixSpan needs to construct a projected database. The main idea of PrefixSpan algorithm (presented in the following paragraph) is to use frequent prefixes to divide the search space and to project sequence databases. Prefix Projected sequential pattern mining method mines the complete set of patterns but greatly reduces the efforts of candidate subsequence generation. Prefix projection substantially reduces the size of projected databases and leads to efficient processing.

How Prefix Works:

Step 1: Find length-1 sequential patterns. Scan database once to find all frequent items in sequences. Each of these frequent items is a length-1 sequential pattern.

Step 2: Divide search space. The complete set of sequential patterns is partitioned according to prefix.

Step 3: Find subsets of sequential patterns. The subsets of sequential patterns can be mined by constructing corresponding projected databases and mine each recursively.

Constraints:

Constraint 1:

(Length constraint) A length constraint specifies the requirement on the length of the patterns, where the length can be either the number of occurrences of items or the number of transactions. Length constraints can also be specified as the number of distinct items, or even the maximal number of items per transactions. For example, a user want to find only long patterns (e.g., patterns consisting of at least 40 visit) in web page sequence analysis. Such a requirement can be expressed by a length constraint len (α) ≡ (len (α) ≥ 40) [12].

Constraint 2:

(Aggregate constraint) an aggregate constraint is the constraint on an aggregate of items in a pattern, where the aggregate function can be sum, avg, max.

Algorithm of PREFIXSPAN: [28]

```
PrefixSpan (α, i, S| α)
Begin
1. Scan S| α once, find the set of frequent items b such that
        b can be assembled to the last element of α to form a sequential pattern; or
        <b> can be appended to α to form a sequential pattern.
2. For each frequent item b, appended it to α to form a sequential pattern α', and output α';
3. For each α', construct α'-projected database S| α', and call
PrefixSpan(α', i+1,S| α').
End
```

Its aim is to search the relevant sequences. The PrefixSpan parameters are a) α which represents a sequential pattern; b) l is the length of α; and c) S| α is the α -projected database if α≠ < >, otherwise, it is the sequence database S.

Example:

Given a set of sequences, where each sequence consists of a list of elements and each element consists of set of items.

< a (abc) (ac) d (cf) > - 5 elements, 9 items

< a (abc) (ac) d (cf) > - 9-sequence

< a (abc) (ac) d (cf) > ≠ <a(ac)(abc)d(cf) >

Table 6. Sample Database

ID	SEQUENCE
10	< a (abc) (ac) d (cf) >
20	< (ad) c (bc) (ae) >
30	< (ef) (ab) (df) cb>
40	< eg (af) cbc >

Min_support=2

PREFIX

- Given two sequences $\alpha = <a_1 a_2 \ldots a_n>$ and $\beta = <b_1 b_2 \ldots b_m>$, $m \leq n$.
- Sequence β is called a prefix of α if and only if:
  - $b_i = a_i$ for $i \leq m-1$;
  - $b_m \subseteq a_m$;
- Example :
  - α =<a(abc)(ac)d(cf)>
  - β =<a(abc)a>

PROJECTION:

- Given sequences α and β, such that β is a subsequence of α.
- A subsequence α' of sequence α is called a projection of α w.r.t. β prefix if and only if:
  - α' has prefix β;
  - There exist no proper super-sequence α'' of α'such that: α'' is a subsequence of α and also has prefix β.

Example:

α =<a(abc)(ac)d(cf)>

β =<(bc)a>

α' =<(bc)(ac)d(cf)>

4.2 FREESPAM:

(Frequent Pattern Projected Sequential Pattern Mining): In FreeSpan uses frequent items to recursively project sequence databases into a set of smaller projected databases and grows subsequence fragments in each projected database [28]. Two alternatives of database projections can be used Level-by-level projection or Alternative-level projection. The method used by FreeSpan divide the data and the set of frequent patterns to be tested, and limits each test being conducted to the corresponding smaller projected database. FreeSpan scan the original database only three times, whatever the maximal length of the sequence.

FreeSpan first scans the database, collects the support for each item, and finds the set of frequent items. Frequent items are listed in support descending order (in the form of item : support) E.g., flist=a:4, b:4, c:4, d:3, e:3, f:3. According to f_list, the complete set of sequential patterns in S can be divided into 6 disjoint subsets: (1) the ones containing only item 'a', (2) the ones containing item 'b', but containing no items after 'b' in flist, (3) the ones containing item 'c', but no items after 'c', in flist, and so on, and finally, (6) ones containing item 'f'. The subsets of sequential patterns can be mined by constructing projected databases. Infrequent items, such as 'g' in this example, are removed from construction of projected databases. Note that {b}, {c}, {d}, {e}, {f}-projected databases are constructed simultaneously during one scan of the original sequence database. All sequential patterns containing only item 'a' are also found in this pass. This process is performed recursively on projected databases. Since FreeSpan projects a large sequence database recursively into a set of small projected sequence databases based on the currently mined frequent sets, the subsequent mining is confined to each projected database relevant to a smaller set of candidates.

4.3 Weighted SPAM:

Weighted sequential pattern mining is also known as Time-Interval Weight Sequential pattern mining (TiWS) [11]. Sequential pattern mining is a topic of data mining concerned with evaluating statistically relevant patterns between data where the values are delivered in a sequence. It is usually presumed that the values are discrete. And thus time series mining is closely related, but usually considered a different activity. Sequence mining is a special case of structured data mining. General sequential pattern mining is based on simple support counting; if weight of the information is used, it finds interesting sequential patterns. For a sequence or a sequential pattern, not only the generation order of data elements but also their time-intervals and generation times are important to get more valuable sequential patterns [30]. Previous sequential pattern mining algorithms use the same priority for each pattern or sequence. The sequential order in a sequence database is important in many situations. In huge datasets, extracting valuable sequential patterns is not easy work. The number of frequent sequential patterns becomes huge as the minimum support becomes lower. So applying weights is effective and efficient to not only generate more important sequential patterns but also adjust the number of sequential patterns. Specifically, it is more effective to apply weight constraints to sequential pattern mining with lower minimum support.

A full periodic pattern is a pattern where every point in time contributes (precisely or approximately) to the cyclic behavior of a time-related sequence. For example, all of the days in the year approximately contribute to the season cycle of the year.

A partial periodic pattern specifies the periodic behavior of a time-related sequence at some but not all of the points in time. For example, Sandy reads the New York Times from 7:00 to 7:30 every weekday morning, but her activities at other times do not have much regularity. Partial periodicity is a looser form of periodicity than full periodicity and occurs more commonly in the real world.

Algorithm [30]

```
Procedure WSpan (WSP, , L, S| )
(1) α is a weighted sequential pattern that satisfies the above pruning conditions,
(2) L is the length of α,
(3 ) S|α is the seq uence d atabase, SDB if α is nu ll, otherwise, it is the α -projected database.
1. Scan S|α once, count the support of each item, and find each weighted frequent item, β in sequences: β is a weighted sequential item if the following pruning condition is not satisfied.
Pruning condition: (support * Max W > min-sup)
(a) β can be assembled to the last element of a to form a sequential pattern or
(b) <β> can be appended to a to form a sequential pattern.
2. For each weighted frequent item β,
Add it to α to form a sequential pattern α', and output α'.
End for
3. For each α',
Construct α'-projected database S|α;
Call WSpan (α', L+1, S|α')
End for
```

Summary of Pattern Growth Based Approaches

Pattern growth methods are much faster than apriori based methods. All the approaches of Pattern growth based methods are described in table 7.

Table 7. Summarized Pattern Growth Based Methods

S. No	Algorithm	Category	Key Feature	Merits/ Demerits
1	PREFIXSPAN	Pattern Growth Based Approach	Use Prefix heuristic and bi level projection	*Merits:* Only 2 passes over data-set "Compresses" data-set No candidate generation Much faster than Apriori *Demerits:* FP-Tree may not fit in memory!! FP-Tree is expensive to build
2	FREESPAN		Pattern Growth based method and use projected sequence database	
3	WEIGHTED SPAN		Time interval based method use	

V. CONCLUSION

Based on an exhaustive literature review this paper presents a study of sequential pattern mining techniques. The taxonomy of sequential pattern mining approaches has given. According to the given taxonomy, sequential pattern mining approaches are divided into two types:

Apriori Based Sequential Pattern Mining

Pattern Growth Based Sequential Pattern Mining

As said in this paper apriori based methods are easy to implement and use large itemset property. But it requires many database scans and also consumes a lot of memory. On the other hand in pattern growth based approaches compressed data is used and only 2 pass over data set. It is much faster than apriori based technique. But it is much expensive to build as compare to apriori based methods.

REFERENCES

[1] Changhai Zhang, Kongfa Hu, Haidong Liu, Youwei Ding & Ling Chen" FMGSP: An Efficient Method of Mining Global Sequential Patterns", FSKD 2007.

[2] Chien-Liang Wu, Jia-Ling Koh, and Pao-Ying An", Improved Sequential Pattern Mining using an Extended Bitmap Representation".

[3] Cláudia Antunes & Arlindo L. Oliveira" Sequential pattern mining with approximated constraints", 2003.

[4] Irfan Khan", PrefixSpan Algorithm Based on Multiple Constraints for Mining Sequential Patterns", International Journal of Computer Science and Management Research Vol 1 Issue 5 December 2012.

[5] Jian Pei, Jiawei Han and Wei Wang", Mining Sequential Patterns with Constraints in Large Databases", CIKM'02, November 4–9, 2002, McLean, Virginia, USA.

[6] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen and Mei-Chun Hsu", Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach", IEEE Transactions On Knowledge And Data Engineering, VOL. 16, NO. 10, OCTOBER 2004.

[7] Jia-Wei Han, Jian Pei and Xi-Feng Yan", From Sequential Pattern Mining to Structure Pattern Mining: A Pattern-Growth Approach", J.Comput.Sci. & Technol. Vol.19, No.3, pp.257-279, May 2004.

[8] Jiawei Han, Hong Cheng, Dong Xin, Xifeng Yan "Frequent pattern mining: current status and future directions" U.S. National Science Foundation NSF, pp.55-86, January 2007

[9] Jian Peis, Jiawei Han & Wei Wang", Mining Sequential Patterns with Constraints in Large Databases", November 4–9, McLean, Virginia, USA, 2002.

[10] Jinhong Li, Bingru Yang & Wei Song", A New Algorithm for Mining Weighted Closed Sequential Pattern", IEEE Second International Symposium on Knowledge Acquisition and Modeling , 2009

[11] J. Mercy Geraldine & G. Shanthi Krishna", Relevant Tiws Pattern Mining With Reduced Search Space", IOSR Journal of Computer Engineering (IOSR-JCE) Vol. 9, Issue 6 ,PP 47-52, 2013

[12] Jyoti Mehta and Rajni Mehta", Prefix Projection: A Technique for Mining Sequential Pattern Included Length and Aggregate", International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11 (2012)

[13] Karam Gouda and Mosab Hassaan", Mining sequential patterns in dense databases", International Journal of Database Management Systems (IJDMS), Vol.3, No.1, February 2011

[14] K.M.V.Madan Kumar, P.V.S.Srinivas & C. Raghavendra Rao", Sequential Pattern Mining With Multiple Minimum Supports by MS-SPADE", International Journal of Computer Science Issues Vol. 9, Issue 5, No 1, , September 2012.

[15] Lionel Vinceslas, Jean-Emile Symphor, Alban Mancheron and Pascal Poncelet", SPAMS: a novel Incremental Approach for Sequential Pattern Mining in Data Streams".

[16] Mahdi Esmaeili and Fazekas Gabor" Finding Sequential Patterns from Large Sequence Data" International Journal of Computer Science Issues, pp. 43-46, Vol.7,Issue 1,No.1,January2010.

[17]Manish Gupta, Jiawei Han "Approaches for pattern discovery Using sequential Data Mining"

[18] Minos N. Garofalakis, Rajeev Rastogi & Kyuseok Shim", Sequential Pattern Mining with Regular Expression Constraints", Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, 1999

[19] Mohammed J. Zaki ", SPADE: An Efficient Algorithm for Mining Frequent Sequences", Kluwer Academic Publishers. Manufactured in The Netherlands, Machine Learning, 42, 31–60, 2001.

[20] NIZAR R. MABROUKEH and C. I. EZEIFE" A Taxonomy of Sequential Pattern Mining Algorithms" ACM Computing Surveys, Vol. 43, No. 1, Article 3, November 2010.

[21] Rajesh Boghey, Shailendra Singh", Sequential Pattern Mining: A Survey on Approaches" IEEE International Conference on CommunicationSystems and Network Technologies, pp.670-674, 2013.

[22] Rakesh Agrawal & Ramakrishnan Srikant" Mining Sequential Patterns" IEEE International Conference, pp.3-14,1995.

[23] Ramakrishnan Srikant & Rakesh." Mining sequential patterns: generalizations and performance improvements", 2003.

[24] Romanas Tumasonis & Gintautas Dzemyda" A probabilistic algorithm for mining frequent sequence", 2001.

[25] Hunor Albert-Lorincz & Jean-Francois Boulicaut" Mining frequent sequential patterns under regular expressions: a highly adaptative strategy for pushing constraints", pp. 316-320, May 1-3, 2003.

[26] Sandra de Amo & Ary dos Santos Rocha Jr." FMGSP: Mining Generalized Sequential Patterns using Genetic Programming", 2005.

[27] Sushila Umesh Ratre & Ravindra Gupta" An Efficient Technique for Sequential Pattern Mining, Volume 3, Issue 3, March 2013

[28] Thabet Slimani and Amor Lazzez", Sequential Mining: Patterns And Algorithms Analysis".

[29] Thomas. Rincy. N & Yogadhar Pandey ", Performance Evaluation on State of the Art Sequential Pattern Mining Algorithms".

[30] Unil Yun, and John J. Leggett", WSpan: Weighted Sequential pattern mining in large sequence databases", 3rd International IEEE Conference Intelligent Systems, September 2006

[31] Vance Chiang-Chi Liao and Ming-Syan Chen", An Efficient Sequential Pattern Mining Algorithm for Motifs with Gap Constraints", IEEE International Conference on Bioinformatics and Biomedicine, 2012.

[32] www.wikipedia.org

[33] Zhenglu Yang and Masaru Kitsuregawa", LAPIN-SPAM: An Improved Algorithm for Mining Sequential Pattern".

[34] Zhigang Zheng, Yanchang Zhao, Ziye Zuo & Longbing Cao", Negative-GSP: An Efficient Method for Mining Negative Sequential Patterns", Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101, 2011