A Survey on Improvements of PrefixSpan Sequential Pattern Mining Algorithm

K. SUNEETHA, M. USHA RANI

1Dept. of MCA, Sri Vidyanikethan Engineering College, Tirupati, Andhra Pradesh, India
2Dept. of Computer Science, Sri Padmavathi Mahila Visvavidyalayam, Tirupati, Andhra Pradesh, India

1umasuni.k@gmail.com, 2musha_rohan@yahoo.com

ABSTRACT

Sequential pattern mining is a significant data-mining method for determining time-related behavior in sequence databases. Discovering sequential patterns is a well-studied area in data mining and has been found in many diverse applications, such as customer purchase behavior analysis, web-log analysis, medical treatments, natural disasters and so on. Most of sequential pattern mining algorithms use a minimum support threshold to prune the combinatorial search space. If the minimum support is low, many spurious patterns having items with different support levels are found; if the minimum support is high, meaningful sequential patterns with low support levels may be missed. In this paper a systematic survey is made on improving the efficiency of traditional sequential pattern mining algorithm called Prefixspan by incorporating various constraints effectively and efficiently into sequential mining process to discover interesting and valuable sequential patterns from sequential databases. At the end, performance analysis is done on the basis of pseudo projection, memory space, number of sequential patterns and execution time supported by various algorithms. Keywords: Prefixspan, Sequential Pattern Mining, constraint-based, pattern-growth approach, projected database.

I INTRODUCTION

Data mining [Chen et al. 1996] is the process of extracting interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from large information repositories such as: relational database, data warehouses, XML repository, etc. Also data mining is known as one of the core processes of Knowledge Discovery in Database (KDD). A Sequential pattern mining[17] an advance of association rule mining, is an imperative subject of data mining, often applied for extracting the useful information [19]. Sequential pattern mining algorithms deals with the problem of determining the frequent sequences in a given database [3] with a user-specified minimum support. A sequence is a collection of an ordered list of item set where item set is a collection of unordered, non empty set of items. Sequential pattern mining is closely related to association rule mining, except that the events are linked by time [20]. A sequential pattern mining is to find all frequent subsequences whose frequency of occurrence is less than minimum support threshold.

Basic Concepts of Sequential Pattern Mining

1. Let \( I = \{x_1, \ldots, x_n\} \) be a set of items, each possibly being associated with a set of attributes, such as value, price, profit, calling distance, period, etc. The value on attribute \( A \) of item \( x \) is denoted by \( x.A \). An itemset is a non-empty subset of items, and an itemset with \( k \) items is called a \( k \)-itemset.
2. A sequence \( a = \langle x_1 \cdots x_l \rangle \) is an ordered list of itemsets. An itemset \( X_i \) (1 \( \leq i \leq l \)) in a sequence is called a transaction, a term originated from analyzing customers’ shopping sequences in a transaction database. A transaction \( X_i \) may have a special attribute, time-stamp, denoted by \( X_i.time \), which registers the time when the transaction was executed. For a sequence \( a = \langle x_1 \cdots x_l \rangle \), we assume \( X_i.time < X_j.time \) for \( 1 \leq i < j \leq l \).
3. The number of transactions in a sequence is called the length of the sequence. A sequence with length \( l \) is called an \( l \)-sequence. For an \( l \)-sequence \( a \), we have len \((a) = l \). Furthermore, the \( i \)-th itemset is denoted by \( a[i] \). An item can occur at most once in an itemset, but can occur multiple times in various itemsets in a sequence.
4. A sequence \( a = \langle x_1 \ldots x_n \rangle \) is called a subsequence of another sequence \( b = \langle y_1 \ldots y_m \rangle \) (\( n \leq m \)), and \( b \) a super-sequence of \( a \), if there exist integers \( 1 \leq i_1 < \ldots < i_n \leq m \) such that \( X_1 Y_{i_1} \ldots X_n Y_{i_n} \).
5. A sequence database SDB is a set of 2-tuples \( \langle sid, a \rangle \), where \( sid \) is a sequence-id and \( a \) a sequence. A tuple \( \langle sid, a \rangle \) in a sequence database SDB is said to contain a sequence \( \gamma \) if \( \gamma \) is a subsequence of \( a \). The number of tuples in a sequence database SDB containing sequence \( \gamma \) is called the support of \( \gamma \), denoted by \( sup(\gamma) \). Given a positive integer \( min.sup \) as the support threshold, a sequence \( \gamma \) is a sequential pattern in sequence database SDB if \( sup(\gamma) \geq min.sup \). The sequential pattern mining problem is to find the complete set of sequential patterns with respect to a given sequence database SDB and a support threshold \( min.sup \).

II CLASSIFICATION OF SEQUENTIAL PATTERN MINING ALGORITHMS

A Classification of existing sequential pattern-mining algorithms is provided in Figure 1. This classification is composed of two main categories of sequential pattern-mining algorithms, namely, apriori-based and pattern-
growth algorithms. According to previous research done in the field of sequential pattern mining, Sequential Pattern Mining Algorithms mainly differ in two ways [15]: (1) The way in which candidate sequences are generated and stored. The main goal here is to minimize the number of candidate sequences generated so as to minimize I/O cost. (2) The way in which support is counted and how candidate sequences are tested for frequency. The key strategy here is to eliminate any database or data structure that has to be maintained all the time for support of counting purposes only. Many algorithms were proposed for sequential pattern mining [2]. These mining methods are classified into two approaches as Apriori-based candidate generation method and pattern growth method. Apriori based approach uses the Apriori principle presented with association mining, which states that any of the super sequence of non frequent pattern cannot be frequent. The major techniques for sequential pattern mining are
1. Apriori Based
2. Pattern Growth Based

a. Apriori-Based Algorithms: The Apriori and AprioriAll [17] set the basis for a breed of algorithms that depend largely on the apriori property and use the Apriori-generate join procedure to generate candidate sequences. The apriori property states that —All nonempty subsets of a frequent itemset must also be frequent. It is also described as ant monotonic (or downward-closed), in that if a sequence cannot pass the minimum support test, its entire super sequences will also fail the test. Apriori based mining techniques such as Apriori-all, GSP [21], SPADE [12], LAPIN-SPAM [22] and LAPIN [23], scan the database multiple times. An ‘n’ size pattern requires n scans of the database and hence these mining techniques are generally inefficient.

b. Pattern-Growth Algorithms: Soon after the apriori-based methods of the mid-1990s, the pattern growth-method emerged in the early 2000s, as a solution to the problem of generate-and-test. The key idea is to avoid the candidate generation step altogether, and to focus the search on a restricted portion of the initial database. The search space partitioning feature plays an important role in pattern-growth. Almost every pattern-growth algorithm starts by building a representation of the database to be mined, then proposes a way to partition the search space, and generates as few candidate sequences as possible by growing on the already mined frequent sequences, and applying the apriori property as the search space is being traversed recursively looking for frequent sequences. FP growth based mining techniques such as Freespan, BIDE [5], COBRA [10], Prefixspan [6], UDDAG [4], etc., utilize a tree based representation that reflects the original database and two scans are required to construct the tree. From this tree, the sequential patterns are derived without reference to the original database. Changes in the original database can easily be reflected in the tree by incremental analysis. These sequential pattern mining algorithms are used for mining Web access patterns from Web logs.

**Figure 1: Classification of Sequential Pattern Mining Algorithms**

**Prefixspan:** Prefixspan is developed by Jian Pei, Jiawei Han and Wei Wang [7] based on the idea of database projection and sequential pattern growth. This algorithm examines only the prefix subsequences after scanning
the sequence database once and then projects their corresponding postfix subsequences into projected database likewise sequential pattern are grown in each projected database by exploring only local frequent sequences. 1. The first step is to scan the sequential database D to get the length-1 sequences. 2. Sequential database is divided into different partitions according to the number of length-1 sequence to get projected databases. 3. Find subsets of sequential patterns; these subsets can be mined by constructing projected databases, and mining each one recursively. Prefixspan algorithm doesn’t generate and tests candidate sequences, nonexistent in a projection database. Projected database keeps on shrinking because only the suffix subsequences of a frequent prefix are projected into a projected database.

The key advantage of Prefixspan is that it does not generate any candidates. It only counts the frequency of local items. It utilizes a divide-and-conquer framework by creating subsets of sequential patterns (i.e. projected databases) that can be further divided when necessary. Its performance is much better than both GSP and Freespan. But the major cost of Prefixspan is the construction of projected databases. To further improve mining efficiency, bi-level projection and pseudo-projection can be used.

To further improve mining efficiency, pseudo and bi-level projections can be used. Pseudo-projection is used when database can be held in the main memory However it is not efficient if the sequential database cannot be held in main memory. A Bi-level projection method is used, which projects databases not at every level but at every two levels. By comparing level-by-level with bi-level projection, bi-level projection reduces the cost of database projection and leads to better performance when the database is large and with low support threshold.

### III EXTENSIONS OF SEQUENTIAL PATTERN MINING TO OTHER TIME-RELATED PATTERNS

In this section, numerous extensions of the initial definition have been studied which may be related to other types of time-related patterns or to the addition of time constraints. Some extensions of these algorithms are multidimensional sequential pattern mining, closed sequential pattern mining, time interval, and constraint based sequential pattern mining

#### i) Multidimensional Sequential Pattern Mining:

Mining sequential patterns with single dimension means that we only consider one attribute along with time stamps in pattern discovery process, while mining sequential patterns with multiple dimensions we can consider multiple attributes at the same time. In contrast to sequential pattern mining in single dimension, mining multiple dimensional sequential patterns introduced by Helen Pinto and Jiawei Han can give us more informative and useful patterns. For example we may get a traditional sequential pattern from the supermarket database that after buying product ‘a’ most people also buy product ‘b’ in a defined time interval. However, using multiple dimensional sequential pattern mining we can further find different groups of people have different purchase patterns.

#### ii) Discovering Constraint Based Sequential Patterns:

Constraint-based mining of sequential patterns is an active research area motivated by many application domains. Basic formulation of the sequential pattern discovery problem assumes that the only constraint to be satisfied by discovered patterns is the minimum support threshold. However, very often users want to restrict the set of patterns to be discovered by adding extra constraints on the structure of patterns. Data mining systems should be able to exploit such constraints in reducing drastically the amount of data to process and speeding-up the extraction time. Jian Pei, Jiawei Han and Wei Wang [8] have systematically presented the problem of pushing various constraints deep into sequential pattern mining using pattern growth methods.

Constraint-based mining may overcome the difficulties of effectiveness and efficiency, since constraints usually represent user’s interest and focus, which limits the patterns to be found to a particular subset satisfying some strong conditions. (Pei, Han, & Wang, 2007) mention seven categories of constraints:

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.
6. **Duration constraint:** A duration constraint is defined only in sequence databases where each transaction in every sequence has a time-stamp. It requires that the sequential patterns in the sequence database must have the property such that the time-stamp difference between the first and the last transactions in a sequential pattern must be longer or shorter than a given period.
7. **Gap constraint:** A gap constraint set is defined only in sequence databases where each transaction in every sequence has a timestamp. It requires that the sequential patterns in the sequence database must have the property such that the timestamp difference between every two adjacent transactions must be longer or shorter than given gap.
In constraint-based sequential pattern mining, the following are other classes of constraints: database constraints, pattern constraints, and time constraints. Database constraints are used to specify the source dataset. Pattern constraints specify which patterns are interesting and should be returned by the query. Finally, time constraints influence the process of checking whether a given data-sequence contains a given pattern.

iii) Closed Sequential Pattern Mining: The sequential pattern mining algorithms developed so far have good performance in databases consisting of short frequent sequences. Unfortunately, when mining long frequent sequences, or when using very low support thresholds, the performance of such algorithms often degrades dramatically. This is not surprising: Assume the database contains only one long frequent sequence \((a_1)(a_2)\ldots(a_{100})\), it will generate \(2^{100}-1\) frequent subsequence if the minimum support is 1, although all of them except the longest one are redundant because they have the same support as that of \((a_1)(a_2)\ldots(a_{100})\). So an alternative but equally powerful solution: instead of mining the complete set of frequent subsequence, we mine frequent closed subsequence only, i.e., those containing no super sequence with the same support. This mining technique will generate a significant less number of discovered sequences than the traditional methods while preserving the same expressive power since the whole set of frequent subsequences together with their supports, can be derived easily from the mining results.

IV IMPROVEMENTS OF PREFIXSPAN ALGORITHM

1. CFML-PREFIXSPAN Algorithm:
An efficient constraint-based sequential pattern mining called CFML-Prefixspan algorithm. This algorithm is devised from the conventional sequential pattern mining algorithm, Prefixspan [7] and used for mining the constraint sequential patterns. In order to discover the most relevant CFML patterns, two concepts namely monetary and compactness are considered along with the frequency and length to the sequential mining process. The goal of constraint-based sequential pattern mining is to determine the entire set of sequential patterns in order to forecast the customer sequence behavior which satisfies the constraint. Aggregate and Duration constraints would be more advantageous in mining sequential patterns from the customer purchasing database. The definition of these two constraints are given below. The CFML-algorithm has utilized Monetary and Compactness constraints that are derived from these two constraints, respectively.

Aggregate constraint: An aggregate constraint [19] describes that the aggregate of items in a sequence should be above or below a given threshold value, which is represented as, \(C_{agg} = \text{Agg}(\alpha) \circ \Delta T\)
where, \(\alpha \in \{\leq, \geq\}\), \(\text{Agg}(\alpha)\) may be sum, average, max, min, standard deviation, and \(\Delta T\) is a given integer.

Duration constraint: A duration constraint [16] describes that the time difference between the first and last items in a sequence should be greater than or less than a predefined threshold value. The duration constraint is represented as, \(C_{dur} = \text{Dur}(\alpha) \circ \Delta T\), where, \(\alpha \in \{\leq, \geq\}\), and \(\Delta T\) is a given integer.

Length Constraint: A length constraint details 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. For instance, a user may desire to find only the long patterns (for example, the patterns consisting of at least 20 transactions) in market-basket analysis. Such a requirement can be expressed by a length constraint, which is defined as, \(C_{len} = (\text{len}(\alpha) \geq 20)\)
Recently, researchers found that Constraint-based sequential pattern mining algorithms [17] have drawn much attention in discovering entire set of sequential patterns by satisfying the constraint.

Initially, the CFML-algorithm mines the 1-length compact frequent patterns (1-CF) by considering the compactness threshold and support threshold. Subsequently, the 1-length compact frequent monetary sequential patterns (1-CFML) are filtered from the mined 1-CF patterns by inputting the monetary constraint. Then, a projected database corresponding to the mined 1-CF patterns is constructed and then the 2-CF patterns are generated using this database. Again, 2-CFML sequential patterns are determined from the 2-CF patterns by integrating the monetary constraint and the process is applied repeatedly until all length constrained-CFML sequential patterns are discovered.

The comparative results of the Prefixspan and CFML-Prefixspan algorithm specify that CFML-Prefixspan algorithm produces lesser number of relevant and profitable sequential patterns than the Prefixspan algorithm. Also it has observed that the computational time required to generate a set of sequential patterns is less than the Prefixspan algorithm for higher threshold values.

2. CIC-PREFIXSPAN Algorithm
Web sequential pattern mining is based on the web access log. Web access log registers the access information of users, including IP address, access time, request URL, referer, user-agent and so on [18][14]. All sequential patterns can be accessed from web logs. But the raw data in the web access log can’t be mined directly. We should preprocess the web access log at first which includes data cleaning, user identification and session identification, and then do the sequential pattern mining with some mining algorithms.

WU et al., [9] proposed the CIC-PrefixSpan, a modified version of PrefixSpan that mines and generates maximal Sequential patterns by combining PrefixSpan and pseudo-projection. First, preprocessing is done to categorize the user sessions into human user sessions, crawler sessions and resource-download user sessions for
efficient Web sequential pattern mining by filtering out the non-human user sessions, leaving the human user sessions and finding the transactions using Maximum Forward Path (MFP). By utilizing CIC-PrefixSpan, the memory space is reduced and generating duplicate projections to find the most frequent path in the users’ access path tree is also avoided. It is shown that CIC-PrefixSpan yields accurate patterns with high efficiency and low execution time compared to GSP and PrefixSpan. However, the frequent substructures within a pattern cannot be mined by CIC-PrefixSpan.

The output of CIC-PrefixSpan algorithm is the complete set of sequential patterns. But in general the number of these sequential patterns is too large. So it is necessary to select the useful and representative ones from all the sequential patterns. Hence maximal sequential patterns are used to represent all the sequential patterns. A sequential pattern $s$ is maximal, if there exists no sequential pattern which is the proper super sequence of $s$. We know that all the subsequences of a sequential pattern are also frequent. So a maximal sequential pattern can represent all its sub-sequential patterns. And it is more useful for the analysis of user behavior. Furthermore, by filtering out the non-maximal sequential patterns, we can reduce the size of the result set of sequential pattern mining. The last step of sequential pattern mining is to find all the maximal sequential patterns. From the experimental results it was observed that, CIC-Prefixspan algorithm significantly improves the mining for longer sequences and for lower support threshold values. Also, complete set of maximal sequential patterns are discovered effectively.

3. **I-PREFIXSPAN Algorithm**: An improved version of Prefixspan named I-Prefixspan [1] proposed by Dhany Saputra [2007]. Its idea is to use sufficient data structure for Seq-Tree framework and separator database to reduce the execution time and memory usage. It improves PrefixSpan in two ways: 1) by implementing ample data structure Seq-Tree framework for in-memory database sequence and constructs index set, and 2) a separator database to store the transaction alteration signs.

Seq-Tree is a general tree with two characteristics: 1) all leaves must be located at the same depth and 2) the height of the tree is at least 2. The ArrayList is used for in-memory sequence database to store items. The ArrayIntList stores the offset for Index set. I-Prefixspan uses separator database to find sequential patterns by comparing the index set of current pattern and the index set of items to be assembled or appended. A separator database helps to reduce the memory space and there is no need to traverse along all items inside all data sequences ie., I-Prefixspan only registers the indices of the transaction time-stamp separators, instead of keeping the in-memory sequence database until the mining is finished.

The experimental results have shown that I-Prefixspan persistently outperforms Prefixspan time-wise and memory-wise. The lower the minimum support, the clearer the excellence performance of I-Prefixspan. When the minimum support is 1%, I-Prefixspan (7.67 seconds) is almost 4 times faster than Prefixspan (30.269 seconds). When the minimum support is dwindled to 0.4%, I-Prefixspan (46.853 seconds) is approximately 4.8 times faster than Prefixspan (225.302 seconds). Moreover, when the support threshold is 0.2%, I-Prefixspan (264.98 seconds) runs almost two orders of magnitude faster than Prefixspan (19,582.97 seconds).

4. **W-PREFIXSPAN Algorithm** For finding weighted frequent sequential patterns, an eminent W-Prefixspan algorithm [11] is developed by modifying the existing traditional Prefixspan algorithm by incorporating two measures such as spending time and recent view into the mining procedure, which uses the pattern growth methodology. From the weighted sequential database, Prefixspan [14] is modified by incorporating the spending time and recent view into the mining procedure. The weightage measure assumed in the proposed W-Prefixspan algorithm is spending time and recent view. The two aspects taken for providing the weightage of the sequential patterns are,

- **Spending time**: One of the fields in the web log data is the duration of the web page which is viewed by the user. Generally, time spent by the user within a particular page is necessary to identify the importance of web pages. From the web page which having long duration, we can conclude that this particular web page has been referred by the user in a long occasion because of its worth. Thus, the spending time is an important measure for the researchers who are attempting to identify the interest of the users. So, if we incorporate the time duration into the mining of sequential patterns, the interesting relationships can be found out from the mined sequential patterns that can be effectively applied to web page recommendation process.

- **Recent view**: Another significant measure taken for sequential pattern mining is recent view that describes whether the page is accessed recently or not. The reason behind taking the recent view for mining the sequential pattern is that the more importance should be given for the web pages which are accessed recently than the older one. The behavior of the user surely vary depend on the time so the recent behavior of the user is significant for finding the sequence analysis. With the intention of behavior variation over time, the recent view is also incorporated into the sequential pattern mining algorithm to achieve a subset of more interesting sequential patterns.

Let us consider a weighted database $ijW$. A sequence $Ws$ is said to be a sub sequence of $Wij$ only if, (1) $W_i$ is a subsequence of $Wij$, $Wij ∈ W_i \{2 \}t_1 < t_2 < \cdots < t_m$ where, $t_i$ is the time at which $Pi_j$ occurred in $Ws$, $1 ≤ r
A sequence is said to be SR sequence \( W_i j \) if and only if, (1) \( W_i \) is a subsequence of \( W_i j \), (2) the \( W \)-support should be satisfied. The following is the procedure for W-Prefixspan algorithm:

- By scanning the weighted sequential database once it finds 1-length weighted sequential patterns.
- It finds 1-SR (Spending time and Recent view) patterns which satisfies the predefined W-Support threshold value.
- Later projection database is formed by projecting the collection of postfixes of mined 1-SR sequence.
- Then, the 2-length SR-patterns are mined from the projected database by computing the weighted support on the projected database.
- This process is repeated recursively until all SR sequential patterns are mined.

**Performance of the W-Prefixspan algorithm**

The performance of the W-Prefixspan algorithm is analyzed with the execution time, memory usage and patterns mined with respect to the existing Prefixspan algorithm. At first, the training data is given to the Prefixspan and W-Prefixspan algorithm to mine the sequential patterns and then, the mining performance of the algorithms are analyzed. The values obtained from the experimentation shows that the execution time of the W-Prefixspan algorithm is 1.5 Sec that is less compared with the time taken of the Prefixspan algorithm.

**CONCLUSION**

In this paper, an Extensions of Sequential Pattern Mining to other time-related patterns are discussed and also a survey is made on improvements of existing traditional Sequential Pattern Mining algorithm called Prefixspan. In addition to frequency some more constraints are incorporated into the traditional p/Prefixspan algorithm, to discover interesting and valuable sequential patterns from sequential databases which can reduce the execution time by more than an order of magnitude, memory space and provides more number of sequential patterns supported by various algorithms.

**REFERENCES**


[5] Jianyong Wang, Jiawei Han, & Chun Li, (2007) “Frequent Closed Sequence Mining without Candidate Maintenance”, IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 8, pp. 1042-1056.


Authors Bibliography

Ms. K. Suneetha obtained Bachelor’s degree in Sciences from S.V.University, Tirupathi and Master’s degree in Computer Applications from S.V.University. She is currently working as Assistant Professor(SL) in the Department of Master of Computer Applications at Sree Vidyanikethan Engineering College, A.Rangampet, Tirupati. She is pursuing her Ph.D. in Computer Science in the area of Data Warehousing and Data Mining. She is in teaching since 2001. She presented many papers at National and International Conferences and published articles in National & International journals.

Dr. M. Usha Rani is an Associate Professor in the Department of Computer Science and HOD for MCA, Sri Padmavati Mahila Visvavidyalayam (SPMVV Woman’s’University), Tirupati. She did her Ph.D. in Computer Science in the area of Artificial Intelligence and Expert Systems. She is in teaching since 1992. She presented many papers at National and International Conferences and published articles in national & international journals. She also has written 4 books like Data Mining - Applications: Opportunities and Challenges, Superficial Overview of Data Mining Tools, Data Warehousing & Data Mining and Intelligent Systems & Communications. She is guiding M.Phil. and Ph.D. in the areas like Artificial Intelligence, Data Warehousing and Data Mining, Computer Networks and Network Security etc.