Constraint-Based Sequential Pattern Mining: A Pattern Growth Algorithm Incorporating Compactness, Length and Monetary

[1]Bhawna Mallick, [1]Deepak Garg, and [2]Preetam Singh Grover
[1]Department of Computer Science & Engineering, Thapar University, India
[2]Department of Computer Science & Engineering, Guru Tegh Bahadur Institute of Technology
GGS Indraprastha University, India

Abstract: *Sequential pattern mining is advantageous for several applications for example, it finds out the sequential purchasing behavior of majority customers from a large number of customer transactions. However, the existing researches in the field of discovering sequential patterns are based on the concept of frequency and presume that the customer purchasing behavior sequences do not fluctuate with change in time, purchasing cost and other parameters. To acclimate the sequential patterns to these changes, constraint are integrated with the traditional sequential pattern mining approach. It is possible to discover more user-centered patterns by integrating certain constraints with the sequential mining process. Thus in this paper, monetary and compactness constraints in addition to frequency and length are included in the sequential mining process for discovering pertinent sequential patterns from sequential databases. Also, a CFML-PrefixSpan algorithm is proposed by integrating these constraints with the original PrefixSpan algorithm, which allows discovering all CFML sequential patterns from the sequential database. The proposed CFML-PrefixSpan algorithm has been validated on synthetic sequential databases. The experimental results ensure that the efficacy of the sequential pattern mining process is further enhanced in view of the fact that the purchasing cost, time duration and length are integrated with the sequential pattern mining process.*

Keywords: *Sequential pattern mining, Constraint-based sequential pattern mining, Constraint, PrefixSpan, Monetary, Compactness.*

Received July 15, 2011; accepted May 22, 2012

1. Introduction

Sequential pattern mining [1, 4, 9, 11, 12, 13, 16, 18, 29], one of the imperative subjects of data mining, is an additional approval of association rule mining [14, 20]. The sequential pattern mining algorithm [2] deals with the problem of determining the frequent sequences in a given database [24]. Sequential pattern mining is sturdily related to association rule mining, excepting that the events of sequential pattern are associated by time [27]. Sequential patterns signify the association among transactions while association rules describe the intra transaction relationships. In association rule mining, the mined output is about the items that are bought together frequently in a single transaction [30]. Whereas, the output of sequential pattern mining represents which items are bought in a particular order by the customers in diverse transactions [31]. Sequential patterns help the managers to find out the items that are bought one after the other in a cycle [13], or to inspect the orders obtained by the browsing of homepages in a Website [22] and more.

In general, the goal of sequential pattern mining algorithms is to discover the sequential patterns from sequential database. Recently, researchers have found that the frequency is not the best measure that can be used to determine the significance of a pattern in different applications. When a single frequency constraint is employed, the conventional mining approaches normally produce a large number of patterns and rules, but majority of them are futile. Due to its ineptitude, the significance of constraint-based pattern mining has increased [15]. In several cases, the user prospects on the discovery process of the mining patterns and the background knowledge of the user have not been considered and so this result in high cost and very hard to deal with the mining process. The sequential pattern mining that handles sequential data (for e.g., the analysis of frequent behaviors) face the same drawbacks. Constraints that limit the number and range of mined patterns are utilized by sequential pattern mining algorithms to reduce this intricacy [3].

In recent times, the constraint-based sequential pattern mining algorithms [23] have drawn much attention among researchers. The goal of constraint-based sequential pattern mining is to determine the entire set of sequential patterns that satisfying a constraint C. A constraint C for sequential pattern mining is a Boolean function C (α) on the set of all sequences [26]. Constraints can be evaluated and distinguished from diverse point of view. Srikant and Agrawal have employed constraint-based sequential pattern mining in their apriori-based GSP algorithm (i.e., Generalized Sequential Patterns,) which generalizes the opportunity of sequential pattern mining by integrating time constraints using sliding time window concept and user-defined taxonomy [28].

In this paper, we have proposed an efficacious constraint-based sequential pattern mining called CFML-PrefixSpan algorithm. The proposed algorithm is devised from the conventional sequential pattern mining algorithm, PrefixSpan [25] and used for mining the constraint sequential patterns. Here, we have considered two concepts namely monetary and compactness that are derived from the aggregate and duration constraints presented in the literature. Initially, the proposed 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 patters (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 paper is structured as follows: Section 2 reviews the recent research works; Section 3 depicts the problem statement; Section 4 describes the proposed algorithm for mining CFML sequential patterns; Section 5 discusses the experimental results; and finally, Section 6 winds up the paper.

2. Review of Related Research

A handful of researches are available in the literature for effective mining of sequential patterns from sequential databases. But recently, most of the researches focus on mining sequential patterns by integrating certain constraints. Some of the recent researches are portrayed here.

A DELISP (delimited sequential pattern) technique has been proposed by Ming-Yen *et al.*[17], which provides the facilities present in the pattern-growth methodology. DELISP has utilized bounded and windowed projection methods to diminish the size of the proposed databases. The time-gap valid subsequences have been maintained by bounded projection and the non-redundant subsequences fulfilling the sliding time-window constraint have been preserved by windowed projection. As well, the delimited growth technique has directly discovered constraint-satisfactory patterns and increased the pace of the pattern growing process. It has been found that the DELISP has excellent scalability and performed better than the eminent GSP algorithm in discovering sequential patterns with time constraints.

The temporal constraints employed for generalized sequential pattern mining have been softened by Celine Fiot *et al.* [10]. Numerous applications necessitate approaches for temporal knowledge discovery. Few of those approaches deal with time constraints among events. Predominantly, some work focuses on extracting generalized sequential patterns. But, such constraints have often been too crisp or required a very accurate assessment to evade flawed information. Hence, an algorithm has been developed on the basis of sequence graphs to manage the temporal constraints while data mining. In addition, as these unstrained constraints may discover more generalized patterns, a temporal accuracy measure has been proposed for supporting the analysis of several mined patterns. For constraint based frequent-pattern mining, Jian Pei *et al.* [26] have designed a framework on the basis of a sequential pattern growth technique. Here, the constraints were effectively pushed deep into the sequential pattern mining under this proposed framework. Also, the framework has been extended to constraint-based structured pattern mining.

Enhong Chen *et al.* [7] have presented robust approaches to cope with tough aggregate constraints. By a theoretical assessment of the tough aggregate constraints on the basis of the concept of total contribution of sequences, two typical types of constraints have been converted into the same form and thus processed in a consistent manner. Subsequently, a PTAC (sequential frequent Patterns mining with Tough Aggregate Constraints) algorithm has been proposed to diminish the cost of using tough aggregate constraints by integrating two efficient approaches. One shuns checking the data items one by one by using the promising features revealed by some other items and validity of the respective prefix. The other evades building a superfluous projected database by successfully eliminating those bleak new patterns, which may otherwise function as new prefixes. Experimental studies performed on the synthetic datasets produced by the IBM sequence generator as well as a real dataset have revealed that the proposed algorithm has gained better performance in speed and space by means of these approaches.

F. Masseglia *et al.* [21] have addressed the problem of mining sequential patterns by handling the time constraints as specified in the GSP algorithm. Sequential patterns were seen as temporal relationships between data present in the database where the considered data was simply the features of individuals or observations of individual behavior. The intent of generalized sequential patterns is to provide the end user with a more flexible handling of the transactions embedded in the database. A proficient GTC (Graph for Time Constraints) algorithm has been proposed to discover such patterns in giant databases. It was based on the idea that handling the time constraints in the initial phase of the data mining process would be highly advantageous. One of the most vital features of the proposed approach is that the handling of time constraint can be easily taken into consideration in conventional level-wise approaches because it is carried out prior to and independently from the counting step of a data sequence. Experiments have shown that the performance of proposed algorithm was substantially faster than the existing sequence mining algorithm.

Yen-Liang Chen *et al.* [8] have defined the RFM sequential pattern and proposed an algorithm for mining all RFM sequential patterns from the customers' purchasing data by integrating the recency, frequency, and monetary (RFM) concept described in the marketing literature. Also, a pattern segmentation framework has been designed by using this algorithm to obtain significant information regarding customer purchasing behavior for managerial decision-making. Experiments have been done on synthetic datasets and a transactional dataset gathered by a retail chain in Taiwan, to analyze the proposed algorithm as well as to empirically expose the benefits of using RFM sequential patterns in examining customers' purchasing data. Moreover, Jigyasa Bisaria *et al.* [6] have proposed a rough set perspective to the problem of constraint driven mining of sequential pattern. The search space of sequential patterns has been partitioned using indiscernibility relation from theory of rough sets and an algorithm has been developed, which allows pre-visualization of patterns and imposition of different kinds of constraints in the mining task. The C-Rough Set Partitioning algorithm was atleast ten times faster than the naïve algorithm SPIRIT that was based on the diverse types of regular expression constraints.

Jigyasa Bisaria *et al.* [5] have investigated the sequential pattern mining problem by two perspectives, one the computational feature of the problem and the other was the integration and adjustability of time constraint. The search space of sequential patterns has been partitioned by indiscernibility relation from theory of rough sets and a robust algorithm has been proposed that allows pre-visualization of patterns and amendment of time constraints before the implementation of mining task. The Rough Set Partitioning algorithm was atleast ten times faster than the naive time constraint based sequential pattern mining algorithm GSP. As well, an extra knowledge regarding the time interval of sequential patterns has been determined using the technique.

3. Problem Statement

The problem of discovering sequential patterns was first introduced in [2] and extended in [28].This section presents a succinct description of sequential pattern mining and constraint based sequential pattern mining. As well, a detailed description of PrefixSpan is given, which is a prominent approach for mining sequential patterns.

3.1. Sequential Pattern Mining

The sequential pattern mining problem is to extract the entire set of sequential patterns with respect to a given sequence database DB and a support threshold $\min\_\sup$.

Let, DB be a sequential database wherein each transaction T holds a customer-id, transaction time, and a set of items involved in the transaction. Let, $I = \{p_1, p_2,,......., p_m\}$ be a set of items. An itemset is a non-empty subset of items, and an itemset with k items is called as k-itemset. A sequence S is an ordered list of itemsets based on their time stamp, which is represented as $< q_1, q_2 \ldots, q_n >$, where $q_i, j \in 1,2 \ldots, n$ is an itemset. A sequence of k items (or of length k) is called as k-sequence. A sequence $< q_1, q_2 \ldots, q_n >$ is a sub-sequence of another sequence $< q_1', q_2' \ldots, q_l' >$, $(n \leq l)$ if there exist integers $i_1 < i_2 < .....i_j..... < i_n$ such as $q_1 \subseteq q_{i_1}', q_2 \subseteq q_{i_2}', \cdots, q_n \subseteq q_{i_n}'$. The mining of sequential patterns is to discover all sequences S such that $\sup(S) \geq \min\_\sup$ for a database DB, given a positive integer $\min\_\sup$ as a minimum support threshold [20, 25].

3.2. Constraint Based Sequential Pattern Mining

The goal of constraint-based sequential pattern mining is to mine the entire set of sequential patterns satisfying a specified constraint C. The literature [26] presents various constraints that are utilized in the sequential pattern mining process. By analyzing all constraints in the literature, it is found that the aggregate and duration constraints would be more advantageous in mining sequential patterns from the customer purchasing database. The definition of these two constrains is given below. The proposed 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,

$$\textbf{1)} \quad C_{agg} \equiv Agg(\alpha)\, \omega\, \Delta T$$

where, $\omega \in \{\leq, \geq\}$, $Agg(\alpha)$ may be sum, average, max, min, standard deviation, and ΔT is a given integer.

Duration constraint: A duration constraint [19] 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,

$$\textbf{2)}\quad C_{dur} \equiv Dur(\alpha)\,\omega\,\Delta T\ ,$$

where, $\omega \in \{\le,\ge\}$ and ΔT is an integer value.

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,

$$\textbf{3)}\quad C_{len} \equiv (len(\alpha) \ge 20$$

3.3. Prefixspan: An Eminent Sequential Pattern Mining Algorithm

PrefixSpan [25] is the most propitious pattern-growth approach, which is based on constructing the patterns recursively. On the basis of Apriori (E.g. GSP algorithm) and pattern growth (E.g. PrefixSpan algorithm) approaches, quite a few algorithms have been proposed for successful sequential pattern mining. Normally, the apriori-like sequential pattern mining approach fall upon some difficulties such as, (i) a large set of candidate sequences could be created in a giant sequence database, (ii) scanning of database multiple times, and (iii) an explosive number of candidates was generated by this apriori-based technique during the time of mining long sequential patterns. In order to overcome such problems, a PrefixSpan algorithm is introduced to effectively discover the sequential patterns. The PrefixSpan algorithm mainly examines the database to identify the frequent 1-sequences. Then, as per these frequent items, the sequence database is projected into different groups, where each group is the projection of the sequence database with respect to the parallel 1-sequence. For these projected databases, the PrefixSpan algorithm continues to find the frequent 1-sequences to form the frequent 2-sequences with the same respective prefix. Repetitively, the PrefixSpan algorithm produces a projected database for all frequent k-sequences to discover the frequent (k+1)-sequences. The basic outline of the PrefixSpan algorithm is given below.

Input: Sequence database D and minimum support threshold $\min\_\sup$

Output: Complete set of sequential patterns.

Method: Call *PrefixSpan* $(\langle\ \rangle, 0, D)$.

Subroutine: *PrefixSpan* $(\alpha, l, \mathrm{D}|_{\alpha})$.

Parameters: α is a sequential pattern; l is the length of α; $\mathrm{D}|_{\alpha}$ is the α-projected database if $\alpha \ne \langle\ \rangle$ (null) otherwise, the sequence database D.

Method:

1. Scan $\mathrm{D}|_{\alpha}$ once and find the set of frequent items f such that,
   a) f can be assembled to the last element of α to generate a sequential pattern or
   b) $\langle f \rangle$ can be affixed to α to generate a sequential pattern.
2. For each frequent item f, append it to α to form a sequential pattern α', and output α'.
3. For each α', create α'-projected database $\mathrm{S}|_{\alpha'}$ and call *PrefixSpan* $(\alpha', l+1, \mathrm{D}|_{\alpha'})$.

4. Proposed Pattern Growth Algorithm by Incorporating Compactness, Monetary and Length Constraints

Sequential pattern mining is the technique of mining sequential patterns whose support is greater than user defined minimal support level. Several researches are available in the literature for discovering the sequential patterns that are mined only based on the concept of frequency. The frequency is an excellent measure for mining the relevant sequential patterns but in real-life problems, frequency alone is not sufficient for finding the user's sequence behavior in any application. Thus, recently, some of the researchers have applied the concept of constraints to discover the most significant patterns in order to forecast the customer sequence behavior. In a supermarket database, the customer behavior in purchasing will not always be static. The customer buying behavior might be changed based on the time and purchasing cost. Accordingly the length of the sequence or the transaction length may also differ. With the aim of facing these challenges in the mining process, we have included three new concepts namely, monetary, length and compactness, into the conventional sequential pattern mining algorithm of our proposed method.

(1) Monetary: Normally, the sequential patterns that occur often in the sequential database are employed to find the significance of the user buying sequences. But in business point of view, there is always a need to consider the cost of an item. This is primarily because there are some patterns that are frequently occurring in the sequential database and are not providing much income. Moreover, the purchasing behavior of the user will be changed based on the cost of an item. For example, the daily required items such as, milk, bread, butter, and cheese are frequently bought by customers, but the valuable goods like gold and diamond are not frequently purchased. It has been though observed the latter items give better profit compared to frequently purchased items.

(2) Compactness: In most practical problems, specifically, pattern learning for managerial decision support, it is vital to include time constraint in the sequential pattern mining task because the customer's purchasing behavior can be varied over time in customer purchasing database. Hence, there is a necessity to consider the time, so that the decision makers who are attempting to find the user sequence behavior can develop better marketing and product strategies. The benefit of compactness is that, it allows drawing out sequential patterns that occur within a reasonable time span. It enables the mining algorithm to provide better solutions for decision makers.

(3) Length: Length constraint for sequential pattern mining is crucial in supermarket data to obtain the interesting patterns. It is well-known that the length is entirely correlated with the time, so including the length constraint into the sequential patterns may result in good decision making in supermarket environment.

In order to discover the most relevant CFML-patterns, we included the concept of monetary and compactness to the sequential mining process along with the frequency and length. The number of purchases made within a certain period, where a higher frequency specifies higher loyalty is called Frequency. Monetary is the amount of cost spent during a certain period, and a higher value discloses that the company should pay more attention to the customer. Compactness defines that the number of purchases made by the customer should be within a reasonable time period. The number of items in a sequence or the number of transactions defines the Length constraint. If the mining process includes the above four concepts, then the decision makers can clearly categorize their customers, and provide a specific score to their customers based on these concepts. As well, the mined patterns can help the company to find out the customers who are more significant.

4.1. CFML-PrefixSpan Algorithm

In this section, we describe an efficient algorithm called CFML-PrefixSpan, which mines all the CFML-patterns from the sequence databases. The CFML-PrefixSpan algorithm is developed by modifying the prominent PrefixSpan algorithm, which exploits the pattern growth methodology for mining the frequent sequential patterns repetitively. We begin by defining the Subsequence, Compact subsequence, Compact Frequent subsequence, Monetary subsequence, and Compact Frequent Monetary subsequence because the proposed CFML-PrefixSpan algorithm utilizes these definitions. Subsequently, we provide a concise description about the proposed CFML-PrefixSpan algorithm.

Let, $S=\langle (p_1,t_1,M_1),(p_2,t_2,M_2),\cdots,(p_n,t_n,M_n)\rangle$ be a data sequence of database D, where p_j is an item, m_j is a purchasing money, and t_j represents the time at which p_j occurs, $1\le j\le n$ and $t_{j-1}\le t_j$ for $2\le j\le n$. P denotes a set of items in the database D.

Definition 1 (Subsequence): A sequence $S_s=\langle (q_1,t_1,M_1),(q_2,t_2,M_2),\cdots,(q_m,t_m,M_m)\rangle$ is said to be a subsequence of S only if, (1) itemset S_s is a subsequence of S, $S_s\in S$ and (2) $t_1<t_2<\cdots<t_m$ where, t_1 is the time at which q_1 occurred in S_s, $1\le r\le m$.

Definition 2 (Length constrained Subsequence): A sequence $S_s=\langle (q_1,t_1,M_1),(q_2,t_2,M_2),\cdots,(q_m,t_m,M_m)\rangle$ is said to be a length constrained subsequence of S only if, (1) itemset S_s is a subsequence of S, $S_s\in S$ and (2) the number of items in S should be equal to l_s.

***Definition 3 (Compact subsequence)**:* Let, $S_s=\langle (q_1,t_1,M_1),(q_2,t_2,M_2),\cdots,(q_m,t_m,M_m)\rangle$ be a sequence of itemsets, where, $t_1<t_2<\cdots<t_m$ and T_C be the predefined compact threshold. S_s is known to be a compact subsequence of S if and only if (1) S_s is a subsequence of S, and (2) the compactness constraint is satisfied, i.e. $t_m-t_1\le Tc$.

Definition 4 (Compact Frequent subsequence) **[19]**: Let, D be a sequential database containing item sets, I and Tc be the predefined compact threshold. S_s is said to be a compact frequent subsequence of D if and only if (1) S_s is a subsequence of D, (2) the compactness constraint is satisfied, i.e. $t_m-t_1\le T_C$, and (3) S_s is a frequent subsequence of database, D.

Definition 5 (Monetary subsequence) **[19]**: Let, $S_s=\langle (q_1,t_1,M_1),(q_2,t_2,M_2),\cdots,(q_m,t_m,M_m)\rangle$ be a sequence of itemsets, where, $t_1<t_2<\cdots<t_m$ and T_m be the predefined monetary threshold. S_s is said to be the monetary subsequence of S if and only if (1) S_s is a

subsequence of S, and (2) the monetary constraint is satisfied, i.e. $\left(\frac{M_1+M_2+\cdots+M_m}{m}\right)\geq T_m$.

Definition 6 (Compact Frequent Monetary Length subsequence): Let, D be a sequential database containing itemsets (I), T_C be the predefined compact threshold, and T_m be the predefined monetary threshold. S_s is said to be a compact frequent monetary subsequence of D if and only if, (1) S_s is a subsequence of D, (2) the compactness constraint is satisfied, i.e. $t_m - t_1 \leq T_C$, (3) S_s is a frequent subsequence of database D, (4) the monetary constraint is satisfied, i.e. $\left(\frac{M_1+M_2+\cdots+M_m}{m}\right)\geq T_m$, and (5) the number of items in S should be equal to l_s.

The important steps involved in the proposed CFML-PrefixSpan algorithm are described below.

```
Input: Sequence database $D$, minimum support
       threshold $min\_sup$, monetary table $M_T$,
       predefined compact threshold $T_C$, and
       predefined monetary threshold $T_m$.
Output: Complete set of CFML-sequential patterns $\beta$.
Method: Call CFML-PrefixSpan $(\langle\ \rangle, 0, D, M_T)$.

Subroutine: CFML-PrefixSpan $(\alpha, l, D|_\alpha, M_T)$
Parameters: $\alpha$ is a sequential pattern; $l$ is the length
       of $\alpha$; $D|_\alpha$ is the $\alpha$-projected database
       if $\alpha \neq \langle\ \rangle$ (null) otherwise, the sequence
       database $D$; $M_T$ is the monetary table.
Method:
1. Scan $D|_\alpha$ once and find the set of compact frequent
   items $f$ such that,
     a) $f$ can be assembled to the last element of $\alpha$ to
        generate a sequential pattern or
     b) $\langle f \rangle$ can be appended to $\alpha$ to generate a
        sequential pattern.
2. For each compact frequent item $f$, append it to $\alpha$
   to form a sequential pattern $\alpha'$.
3. For each $\alpha'$,
     a) Check monetary using $M_T$.
     b) Check length threshold $l_s$.
4. Create a set $\beta$ from $\alpha'$ by substituting the findings
   of step 3.
5. For each $\alpha'$, create $\alpha'$-projected database $D|_{\alpha'}$, and
   call PrefixSpan ($\alpha', l+1, D|_{\alpha'}, M_T$).
```

Step 1: Finding 1-CFML patterns: Originally, the sequential database D and monetary table M_T are given to the proposed CFML-Prefix Span algorithm. Then, the 1- CFML sequential patterns are mined from the sequential database by scanning the database once. The 1-CF patterns (compact frequent) that satisfy the predefined compact threshold and support threshold are mined from the sequential database by simply scanning the database. Subsequently, the monetary constraint is applied on the 1-CF patterns, so that we can obtain a set of 1- CFML patterns.

Example 1: Let, D be the sequential database given in Table 1 and M_T be the monetary table given in Table 2. We scan the database once and find the set of items that satisfy the predefined compact threshold ($Tc = 4$) and predefined support ($min\_sup = 2$): [(shampoo $\rightarrow 2$), (toothpaste $\rightarrow 3$), (soap $\rightarrow 3$), (hair oil $\rightarrow 2$), (perfume $\rightarrow 1$))]. In this set, the patterns that satisfy the compact threshold and support threshold are called as 1-CF patterns, they are [(shampoo $\rightarrow 2$), (toothpaste $\rightarrow 3$), (soap $\rightarrow 3$), (hair oil $\rightarrow 2$)]. Then, we compute the monetary of the 1-CF patterns obtained from the previous step, [shampoo $\rightarrow$ (2, 5), toothpaste $\rightarrow$ (3, 10), soap $\rightarrow$ (3, 15), hair oil $\rightarrow$ (2, 2)]. Based on the monetary threshold ($T_m = 10$), we obtain a set of 1-CFML patterns as {toothpaste and soap}.

Table 1. Sequential database.

Customer Id	Sequence
A	<(shampoo,1), (toothpaste,3), (soap,4), (hair oil,4), (perfume,5)>
B	<(toothpaste,1), (soap,2), (hair oil,3)>
C	<(shampoo,3), (toothpaste,4), (soap,4)>

Table 2. Monetary table.

Item	Monetary Value
Shampoo	5
Toothpaste	10
Soap	15
Hair Oil	2
Perfume	10

Step 2: Dividing search space: The mined 1-CF patterns are then employed to create a projected database, which is the collection of postfixes of sequence with regard to the prefix (1-CF pattern).

Suppose, if the projection set contains k number of patterns, then k disjoint subsets can be obtained from the sequential database using the whole set of 1-length compact frequent patterns.

Example 2: Here, we create a projected database for the 1-CF patterns such as {shampoo, toothpaste, soap and hair oil}. The steps followed for creating the projected database of the pattern <shampoo> are: by looking at the first sequence in the database, <shampoo> has a time stamp value of 1. Thus, the projection based on the first sequence is obtained by taking the postfixes of pattern <shampoo> (sequences after the time stamp 1) in the first sequence. Likewise, we obtain the projection for the remaining sequences present in the sequential database. The projected database for the pattern <shampoo> contains < (toothpaste, 3), (soap, 4), (hair oil, 4), (perfume, 5)> and < (toothpaste, 4), (soap, 4) >. In the same way, the projection is done for other 1-CF patterns. Table 3 illustrates the projected database of all one length CF patterns in the projection set.

Step 3: Finding subsets of sequential patterns: Here, a set of 2-length compact frequent patterns are mined by scanning the projected database once. Subsequently, a set of 2-CF patterns are obtained by applying the monetary constraint on the 2-length compact frequent patterns. Again, the projected database is created using the mined 2-CF patterns and this process is repeated recursively until all CFML patterns are determined for the given threshold l_s .

Table 3 .Projected database for 1-length Compact Frequent pattern.

Item	Sequential Pattern
<shampoo>	<(toothpaste,3), (soap,4), (hair oil,4), (perfume,5)> < (toothpaste,4), (soap,2)>
<toothpaste>	<(soap,4), (hair oil,4), (perfume,5)> <(soap,2), (hair oil,3)> < (soap,4)>
<soap>	<(hair oil,4), (perfume,5)> <(hair oil,3)>
<hair oil>	<(perfume,5)>

Example 3: The projected database created by the 1-length CF sequential patterns is employed for mining all 2-length CFML sequential patterns. The steps followed for mining the 2-length CFML sequential patterns having prefix $\langle shampoo \rangle$ are: Initially, we obtain the count of the compact frequent items by scanning the projected database once, which is represented as, [(shampoo $\rightarrow$ 0), (toothpaste $\rightarrow$ 2), (soap $\rightarrow$ 1), (hair oil $\rightarrow$ 0), (perfume $\rightarrow$ 0)]. In the above set, the pattern that satisfies the compact threshold and support threshold is [(toothpaste $\rightarrow$ 2)]. The mined 2-CF sequential pattern is {shampoo toothpaste}. Subsequently, we apply the monetary constraint on the mined 2-CF sequential pattern so that, we can obtain the 2-CFML pattern [<shampoo toothpaste> $\rightarrow$ (2, 7.5)]. Here, there is no 2-length CFML sequential pattern having a prefix <shampoo> because the pattern <shampoo toothpaste> has not satisfied the given monetary threshold. Again, we create a projected database based on the 2-CF sequential patterns and the 3-CF patterns are obtained by scanning the projected database. Then, we mine all length CFML patterns with prefix <shampoo> recursively. The aforesaid procedure is repeated for other 1-CF patterns such as, <toothpaste>, <soap> and <hair oil>. The mined CFML-patterns are {<toothpaste>, <toothpaste soap>, <soap>} for the threshold length $l_s = 2$.

5. Experimental Results and Performance Analysis

The experimental results of the proposed CFML-PrefixSpan algorithm for efficacious mining of CFML patterns is described in this section. The proposed CFML-PrefixSpan algorithm is programmed by means of JAVA (jdk 1.6). The sample database taken for experimentation is given in Table 1 and the monetary constraint is given in Table 2. Such database and monetary table are given as an input to the proposed CFML-PrefixSpan algorithm for successful mining of CFML sequential patterns. Originally, we mined the 1-CFML patterns based on the thresholds, $T_C = 4$, $\min\_sup = 2$, $T_m = 10$, $l_s = 2$. Subsequently, the projection was done based on the mined 1-length compact frequent patterns. The projected database for the 1-CF pattern is shown in the Table 3. Eventually, we obtained a complete set of CFML patterns for the given input sequential database. The obtained complete set of CFML patterns is {<toothpaste>, <toothpaste soap>, <soap>}.

The comparative results of the PrefixSpan with our proposed CFML-PrefixSpan algorithm are given in Table 4. It clearly ensures that the proposed algorithm provides lesser number of sequential patterns than the PrefixSpan algorithm. The PrefixSpan algorithm contains less profitable and longer time length sequential patterns (<shampoo>, <shampoo toothpaste>, <shampoo soap>, <hair oil> and <toothpaste hair oil>) whereas, the proposed algorithm generates only the profitable and relevant CFML sequential patterns (<toothpaste>, <toothpaste soap> and <soap>). Thus, from the business point of view, the CFML-PrefixSpan algorithm is more applicable for developing better business strategies than the PrefixSpan algorithm.

Table 4. Comparison of the proposed algorithm with PrefixSpan algorithm.

Item	CFML-Sequential Pattern	Sequential Pattern
<shampoo>		<shampoo>, <shampoo toothpaste>,<shampoo soap>
<toothpaste>	<toothpaste>, <toothpaste soap>	<toothpaste>, <toothpaste soap>, <toothpaste hair oil>
<soap>	<soap>	<soap>
<hair oil>		<hair oil>

5.1. Performance Analysis for Various Length-Threshold

In order to evaluate the performance of proposed algorithm, a synthetic dataset has been employed. Here, we have created a sequential database that holds 10,000 sequences of 10 items. The synthetic dataset is given to the proposed CFML-PrefixSpan algorithm for mining the CFML sequential patterns. The predefined threshold values given to our algorithm are, $T_C = 4$, $\min\_sup = 1000$ and $T_m = 10$. Based on the given database and other parameters, the algorithm produced a complete set of CFML sequential patterns for the given length-threshold l_s. Then, the same sequential database is given to the PrefixSpan algorithm for mining the sequential patterns. The results obtained from both the algorithms are shown in Table 5 and the plotted graph is illustrated in Figure 1. Figure 1 proves that less number of patterns are generated by the proposed algorithm than the PrefixSpan algorithm.

Table 5. Number of sequential patterns generated by PrefixSpan and CFML-PrefixSpan algorithm.

Length Threshold	Number of CFML Sequential Patterns	Number of Sequential Patterns
$l_s = 1$	8	10
$l_s = 2$	143	200
$l_s = 3$	24	1000

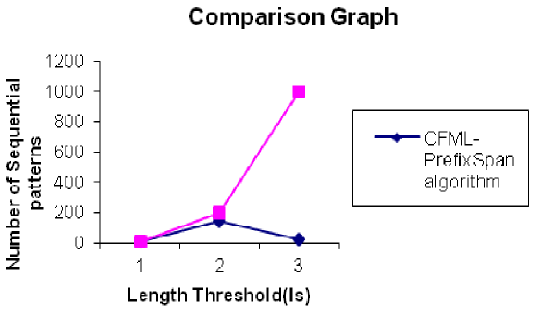

Figure 1. Comparison graph between PrefixSpan and CFML- PrefixSpan algorithm.

Then, the computation time is considered, one of the important parameters to find the intricacy of the algorithm. Initially, by inputting the $\min\_sup = 1000$ and $T_m = 10$, we discover a set of sequential patterns such a way the time taken by the algorithms are obtained for various threshold T_C (for CFML-PrefixSpan) and length (PrefixSpan). The time required to complete the mining task is computed and the values are plotted in a graph, which is shown in figure 2. While comparing the computational complexity, it has been found that the proposed algorithm has taken less computation time than the PrefixSpan algorithm for higher threshold values.

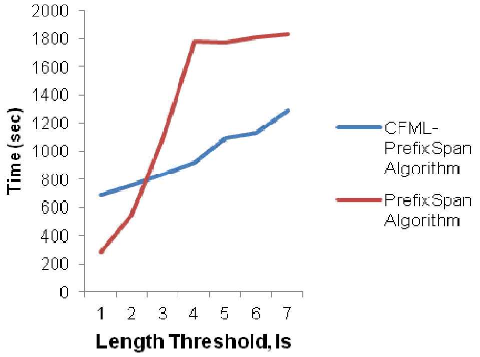

Figure2. Run time performance of the algorithm.

5.2. Performance Evaluation for Various Compact Thresholds

For performance comparison, the synthetic datasets are given to both the proposed algorithm as well as PrefixSpan algorithm in order to discover a set of sequential patterns. These two algorithms are compared in terms of the number of useful sequential patterns obtained for diverse support thresholds. By inputting the $\min\_sup = 1000$ and $T_m = 10$, the results are computed by varying the T_C and the obtained results are shown in the Figure 3. From the graph, it is clear that the sequential patterns obtained by the proposed algorithm are considerably less compared to the PrefixSpan algorithm because the proposed algorithm is capable of discovering more relevant sequential patterns.

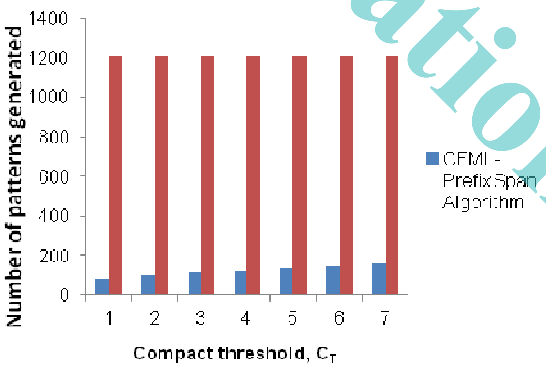

Figure3. Comparison graph of the $\min\_sup = 1000$ and $T_m = 10$.

6. Conclusions

We have presented a robust CFML-PrefixSpan algorithm for mining all CFML sequential patterns from the customer transaction database. The CFML-PrefixSpan algorithm has utilized a pattern-growth methodology that discovers sequential patterns via a divide-and-conquer strategy. Here, we have mainly applied two innovative concepts namely, monetary and compactness that are derived from the aggregate and duration constraints in addition to frequency for mining the most interesting sequential patterns. In our algorithm, the sequence database was recursively projected into a set of smaller projected databases based on the compact frequent patterns. As well, CF-sequential patterns were determined from each projected database by exploring only the locally compact frequent items and then, the CFML sequential patterns were discovered. The mined CFML sequential patterns has provided the valuable information regarding the customer purchasing behavior and ensure that all patterns have reasonable time spans with good profit. The experimental results have confirmed that the potency of sequential pattern mining algorithms can be improved substantially by integrating the monetary and compactness concepts into the mining process.

Acknowledgement

The authors would like to thank the Editor and anonymous referees for providing valuable comments and constructive suggestions. We express our gratitude towards the Galgotias College of Engineering and Technology, Greater Noida for the support being provided to complete this research work.

References

[1] Agrawal R., Imielinski T., and Swami A., "Database Mining: A Performance Perspective," *IEEE Transaction Knowledge and Data Engineering*, vol. 5, no. 6, pp. 914-925, 1993.

[2] Agrawal R. and Srikant R., "Mining Sequential Patterns," *in Proceedings of the 11th International Conference on Data Engineering*, pp. 3-14, Taipei, Taiwan, 1995.

[3] Antunes C. and Oliveira A.L., "Sequential Pattern Mining With Approximated Constraints," *in proceedings of the International Conference on Applied Computing*, pp. 131-138, 2004.

[4] Bigus J., "Data Mining with Neural Networks: Solving Business Problems From Application Development To Decision Support," *McGraw-Hill*, 1996, ISBN: 0-07-005779-6.

[5] Bisaria J., Shrivastava N., and Pardasani K.R., "A Rough Sets Partitioning Model for Mining Sequential Patterns with Time Constraint," *International Journal of Computer Science and Information Security*, vol. 2, no. 1, pp. 1-9, June 2009.

[6] Bisaria J., Srivastav N., and Pardasani K.R., "A Rough Set Model for Sequential Pattern Mining with Constraints," *in Proceedings of the (IJCNS) International Journal of Computer and Network Security*, vol. 1, no. 2, November 2009.

[7] Chen E., Cao H., Li Q., and Qian T., "Efficient Strategies for Tough Aggregate Constraint-Based Sequential Pattern Mining," *Information Sciences*, vol. 178, no. 6, pp. 1498-1518, March 2008.

[8] Chen Y., Kuo M., Wu S., and Tang K., "Discovering Recency, Frequency, and Monetary (RFM) Sequential Patterns from customers' Purchasing Data," *Electronic Commerce Research and Applications*, vol. 8, no. 5, pp. 241-251, 2009.

[9] Fayyad U., Shapiro G., and Smyth P., "From Data Mining to Knowledge Discovery: An Overview," *Advances in Knowledge Discovery and Data Mining*, AAAI Press/The MIT Press, 1996.

[10] Fiot C., Laurent A., and Teisseire M., "Extended Time Constraints for Sequence Mining," *in Proceedings of the 14th International Symposium on Temporal Representation and Reasoning(TIME'07)*, pp. 105-116, June 28-June 30 , Alicante, Spain, 2007.

[11] Frawley W., Shapiro G., and Matheus C., "Knowledge Discovery in Databases: An Overview," *AI Magazine*, fall 1992, vol. 13, no. 3, pp. 213-228, 1992.

[12] Han J. and Fu Y., "Attribute-Oriented Induction in Data Mining," *Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press*, pp. 399- 421, 1996.

[13] Han J. and Kamber M., "Data Mining: Concepts and Techniques," *Morgan Kaufman publishers*, 2001, ISBN: 1-55860489-8.

[14] Hou S. Zhang X., "Alarms Association Rules Based on Sequential Pattern Mining Algorithm," *in proceedings of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 2, pp. 556-560, Shandong, 2008.

[15] Hu Y., "The Research of Customer Purchase Behavior Using Constraint-based Sequential Pattern Mining Approach," *Thesis Report*, 2007.

[16] Julisch K., "Data Mining for Intrusion Detection -A Critical Review," *Application of Data Mining In Computer Security*, Kluwer Academic Publisher, Boston, 2002.

[17] Lin M. and Lee S., "Efficient Mining of Sequential Patterns with Time Constraints by Delimited Pattern Growth," *Knowledge and Information Systems*, vol.7, no. 4, pp. 499-514, 2005.

[18] Mallick B., Garg D., and Grover P.S., "Incremental Mining of Sequential Patterns: Progress and Challenges," *accepted by Intelligent Data Analysis*, ISSN 1088-467X.

[19] Mallick B., Garg D., and Grover P.S., "CFM-PrefixSpan: A Pattern Growth Algorithm Incorporating Compactness and Monetary," *International Journal of Innovative Computing, Information and Control*, ISSN 1349-4198, vol. 8, no. 7(A), pp. 4509-4520, July 2012.

[20] Masseglia F., Poncelet P., and Teisseire M., "Incremental Mining of Sequential Patterns in Large Databases," *Data & Knowledge Engineering*, vol. 46, no.1, pp. 97-121, 2003.

[21] Masseglia F., Poncelet P., and Teisseire M., "Efficient Mining of Sequential Patterns with Time Constraints: Reducing the Combinations," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2677-2690, March 2009.

[22] Myra S., "Web Usage Mining for Web Site Evaluation," *Communications of the ACM*, vol. 43, no. 8, pp. 127-134, 2000.

[23] Orlando S., Perego R., and Silvestri C., "A New Algorithm for Gap Constrained Sequence Mining," *In Proceedings of the ACM Symposium on Applied Computing*, pp. 540 – 547, Nicosia, Cyprus, 2004.

[24] Parmar J.D. and Garg S., "Modified Web Access Pattern (mWAP) Approach for Sequential Pattern Mining," *Journal of computer Science*, vol. 6, no.2, pp. 46-54, June 2007.

[25] Pei J., Han J., Asl B.M., Wang J., Pinto H., Chen Q., Dayal U., and Hsu M., "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach," *IEEE transactions on knowledge and data engineering*, vol. 16, no. 10, October 2004.

[26] Pei J., Han J., and Wang W., "Constraint-Based Sequential Pattern Mining: the Pattern-Growth Methods," *Journal of Intelligent Information Systems*, vol. 28, no. 2, pp. 133-160, 2007.

[27] Sobh T., "Innovations and Advanced Techniques in Computer and Information Sciences," *Springer*, ISBN 978-1-4020-6268-1, 2007.

[28] Srikant R. and Agrawal R., "Mining Sequential Patterns: Generalizations and Performance Improvements," *in Proc. of the 5th International Conference on Extending Database Technology (EDBT'96)*, pp. 3-17, Avignon, France, September 1996.

[29] Tang H., Fang W., and Cao Y., "A Simple Method of Classification with VCL Components," *in Proceedings of the 21st international CODATA Conference*, 2008.

[30] Yafi E, Al-Hegami A. S., Alam Afsar, and Biswas Ranjit, "YAMI: Incremental Mining of Interesting Association Patterns," *The International Arab Journal of Information Technology*, vol. 9, no. 6, November 2012.

[31] Zhao Q. and Bhowmick S.S., "Sequential Pattern Mining: A Survey," *Technical Report, CAIS, Nanyang Technological University*, Singapore, no.118, 2003.

Bhawna Mallick received the B.Tech in Computer Technology from Nagpur University, Nagpur, India and M.Tech in Information Technology from Punjabi University, Patiala, India. She is currently pursuing the Ph.D. degree and working as Head, Department of Computer Science & Engineering at Galgotias College of Engineering & Technology, Greater Noida affiliated to UP Technical University, India. She has 13 years of industry and academic experience with organizations like Infosys Technologies Ltd, Chandigarh, India and NIIT Technologies Ltd, New Delhi, India. She is member of IEEE (Institute of Electrical and Electronics Engineers), USA. Her area of research is Data Mining focusing on Sequential Mining of Progressive Databases.

Deepak Garg has done his Ph.D. in the area of efficient algorithm design from Thapar University. He is certified on latest technologies from Sun for Java Products, IBM for Web services and Brain bench for Programming concepts. He is Senior Member of IEEE (Institute of Electrical and Electronics Engineers), USA, Executive Member of IEEE Delhi Section and secretary of IEEE Computer Society, Delhi Section. He is Life Member of ISTE, CSI, IETE (Institute of Electronics and telecommunication Engineers), ISC (Indian Science Congress), British Computer Society and ACM, UK. He started his career as a Software Engineer in IBM Corporation Southbury, CT, USA and then with IBM Global Services India Pvt Ltd, Bangalore, India. He is presently working as Professor, Thapar University, Patiala.Deepak has 37 publications in International Journals and conferences. He is on the Editorial Board of seven International Journals. His active research area is Data Structure, Algorithms and Data Mining.

Preetam Singh Grover received his Master's degree and doctorate from Delhi University, Delhi, India. He is widely travelled and delivered invited talks/key note addresses at many National/International Conferences/Seminars and Workshops. He is on the Editorial Board of four International Journals. He has written 9 books and many of his articles have appeared in several books published by IEEE of USA. He has published more than 100 research papers in international and national journals and conferences including published by IEEE, ACM and Springer. He is presently Director General at Guru Tegh Bahadur Institute of Technology, GGS Indraprastha University, Delhi, India. Formerly he was Dean & Head of Computer Science Department, Delhi University, Delhi, India. Prof Grover is a member of IEEE Computer Society. His current research interests are: Component based and Aspect-oriented Software Engineering, Autonomic Embedded Systems.