International Journal of Innovative Computing, Information and Control
Volume **8**, Number **7(A)**, July **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 4509–4520

CFM-PREFIXSPAN: A PATTERN GROWTH ALGORITHM INCORPORATING COMPACTNESS AND MONETARY

Bhawna Mallick[1,3], Deepak Garg[1] and P. S. Grover[2]

[1]Department of Computer Science and Engineering
Thapar University
P.O. Box 32, Patiala, Pin 147004, India
bhawnamallickphd@gmail.com

[2]Guru Tegh Bahadur Institute of Technology
GGS Indraprastha University
G-8 Area, Rajouri Garden, New Delhi - 11006, India

[3]Department of Computer Science and Engineering
Galgotias College of Engineering and Technology
1, Knowledge Park, Phase II, Greater Noida 201306, India

Received March 2011; revised July 2011

Abstract. *Prior researches in the field of mining sequential patterns are based on the concept of frequency and assume that the customer purchasing behavior sequences do not vary over time/purchasing money. To adapt the sequential patterns to these changes, two concepts, namely monetary and compactness, are incorporated with the conventional sequential pattern mining technique. Incorporating specific constraints with the sequential mining process has enabled the discovery of more user-centered patterns. In this paper, we incorporate two constraints, namely monetary and compactness in addition to frequency with the sequential mining process for discovering remarkable and helpful sequential patterns from sequential databases. For incorporating the constraints with the original PrefixSpan algorithm, we have presented a CFM-PrefixSpan algorithm for mining all CFM sequential patterns from the sequential database. The proposed CFM-PrefixSpan algorithm has been validated on synthetic and real sequential databases. The experimental results ensure the effectiveness of the discovered sequential patterns since purchasing money and time length were incorporated with the sequential pattern mining process.*

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

1. **Introduction.** Sequential pattern mining is a vital subject of data mining, an additional endorsement of association rule mining, and it is also extensively applied [11,13]. Sequential pattern mining algorithms [16] deal with the problem of finding out the existing frequent sequences in a given database [14]. Sequential pattern mining is very similar to association rule mining, with the difference that the events of sequential pattern are associated by time [12]. Sequential patterns specify the association among transactions whereas association rules characterize intra transaction relationships. In association rule mining, the mined output is about the items which are brought together very often in a single transaction, whereas the output of sequential pattern mining is about which items are bought in a particular order by the same customer in different transactions [9]. Sequential patterns can assist managers to determine which items are bought one after the other in a cycle, or to examine orders obtained by the browsing of homepages in a Web site [17] and more.

Recently, researchers have acknowledged that frequency is not the best measure that can be employed to decide the importance of a pattern in different applications. When a single frequency constraint is employed, the traditional mining techniques commonly produce a large number of patterns and rules, but the majority of them are of no use. As a result of its inefficiency and ineffectiveness, the significance of constraint-based pattern mining has increased [20]. In many cases, there is failure to focus from the user prospect on the discovery process of the mining patterns and on the background knowledge of the user. This led to a highly expensive and very hard to deal procedure. The sequential pattern mining that handles sequential data (for example, the analysis of frequent behaviors) experiences the same drawbacks. Constraints that limit the number and range of discovered patterns are used by sequential pattern mining algorithms to reduce this difficulty [19].

In recent times, constraint-based sequential pattern mining algorithms [15] have received a great deal of attention among researchers. The problem of constraint-based sequential pattern mining is to identify the entire set of sequential patterns that satisfy a specified constraint C. A constraint C for sequential pattern mining is a Boolean function C (α) on the set of all sequences [10]. Constraints can be analyzed and distinguished from diverse point of views. The constraint categories include time constraints, item constraints, length constraints, super-pattern constraints, regular expression constraints, and user defined constraints and so on. Srikant and Agrawal have utilized constraint-based sequential pattern mining in their apriori-based, improved algorithm GSP (i.e., Generalized Sequential Patterns). This algorithm generalizes the opportunity of sequential pattern mining by adding user-defined taxonomy and time constraint using sliding time window concept [18]. A handful of researches are available in the literature for effectual mining of constraint sequential patterns from sequential databases. A concise review of some of the recent researches is given in [1-6,10].

In this paper, we have developed an efficient constraint-based sequential pattern mining, known as CFM-PrefixSpan algorithm. The proposed algorithm is tailored from the traditional sequential pattern mining algorithm, PrefixSpan [8]. Here, we have used two concepts namely, monetary and compactness that are derived from the aggregate and duration constraints which are presented in the literature. At first, the proposed algorithm mines the 1-length compact frequent patterns (1-CF) by considering the compactness threshold and support threshold. Then, we filter the 1-length compact frequent monetary sequential patters (1-CFM) from the mined 1-CF patterns by inputting the monetary constraint. Subsequently, we build the projected database corresponding to the mined 1-CF patterns and then use it to generate the 2-CF patterns. Again, we find the 2-CFM sequential patterns from it by incorporating the monetary constraint and the process is applied recursively until all length CFM sequential patterns are mined.

The basic outline of the paper is as follows. The problem statement is described in Section 2 and the proposed algorithm for mining CFM sequential patterns is given in Section 3. The experimental results and its discussion are presented in Section 4. Conclusion is summed up in Section 5.

2. Problem Statement.

The problem of mining sequential patterns was first introduced in [16] and extended in [18]. This section presents a concise description of sequential pattern mining and constrained sequential pattern mining. In addition, a detailed description of PrefixSpan, which is an eminent method for mining sequential patterns, is given for the completeness of this article.

2.1. **Sequential pattern mining.** The sequential pattern mining problem is to mine the complete set of sequential patterns with respect to a given sequence database D and a support threshold min_sup.

Let D be a sequential database where each transaction T contains customer-id, a transaction time and a set of items entailed in the transaction. Let $I = \{p_1, p_2, \ldots, p_m\}$ be a set of items. An itemset is a non-empty subset of items, and an itemset with k items is called a k-itemset. A sequence S is an ordered list of itemsets based on their time stamp. It is represented by $< q_1, q_2 \ldots, q_n >$, where q_i, $j \in 1, 2 \ldots, n$ is an itemset. A sequence of k items (or of length k) is called 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, $i_1 < i_2 < \ldots i_j \ldots < i_n$ 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 D, given a positive integer min_sup as a minimum support threshold [8,13].

2.2. **Constrained sequential pattern mining.** The problem of mining constraint-based sequential patterns is to discover the complete set of sequential patterns satisfying a specified constraint C. The literature [10] presents several constraints that are used in the sequential pattern mining process. By examining all the constraints in the literature, we observed that the use of aggregate and duration constraint to mine sequential patterns from the customer purchasing database would be more preferable and effective. The definition of these two constrains is given below. The proposed algorithm has used the monetary and compactness constraint that are derived from these two constraints respectively.

Constraint 1 (Aggregate constraint): An *aggregate constraint* defines that the aggregate of items in a sequence must be longer than or shorter than a given threshold value. It is formally represented as

$$C_{agg} \equiv Agg(\alpha)\omega\Delta T$$

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

Constraint 2 (Duration constraint): A duration constraint defines that the time difference between the first and last items in a sequence must be greater than or less than a predefined threshold value. A duration constraint is represented in the following form,

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

where, $\omega \in \{\leq, \geq\}$ and ΔT is an integer value. A sequence α satisfies the duration constraint if and only if $\left| \left\{ \begin{array}{l} \beta \in SDB \,|\, \exists\, 1 \leq i_1 < \cdots < i_{len(\alpha)} \leq len(\beta) \text{ s.t.} \\ \alpha[1] \subseteq \beta[i_1], \cdots, \alpha[len(\alpha)] \subseteq \beta\left[i_{len(\alpha)}\right] \text{ and} \\ \left(\beta\left[i_{len(\alpha)}\right].\text{time} - \beta[i_1].\text{time}\right)\omega\Delta T \end{array} \right\} \right| \geq$ min_sup.

2.3. **Prefixspan: An eminent sequential pattern mining algorithm.** PrefixSpan [8] is the most capable of the pattern-growth techniques and it is based on constructing patterns recursively. On the basis of Apriori (e.g., GSP algorithm) and pattern growth (e.g., PrefixSpan algorithm) techniques, quite a few algorithms have been developed for the efficient sequential pattern mining. Generally, the apriori-like sequential pattern mining technique has come across a lot of difficulties such as, (a) in a huge sequence database, a large set of candidate sequences could be developed, (b) it involves multiple scans of databases in mining and (c) an explosive quantity of candidates was produced by the apriori-based technique for long sequential patterns. So as, to resolve these problems, PrefixSpan algorithm is introduced to mine the sequential patterns. The PrefixSpan algorithm primarily examines the database to locate 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 corresponding same prefix. Recursively, the PrefixSpan algorithm produces a projected database for every frequent k-sequence to locate the frequent $(k + 1)$-sequences.

3. **Proposed Pattern Growth Algorithm by Incorporating Compactness and Monetary Constraints.** Several researches are available in the literature for mining the sequential patterns that are mined only based on the concept of frequency. Though the frequency is a good measure for mining the effectual sequential patterns but usually in real-life problems, frequency alone is not that efficient for finding the user's sequence behavior in any application. Therefore, recently, some of the researchers have used the concept of constraints to find the relevant and useful patterns to predict the customer sequence behavior. In a supermarket database, the customer behavior of buying sequence is not always a static one; it will be a dynamic environment. Hence, the customer buying behavior might be changed based on time and purchasing money. With the aim of adapting to these challenges in the mining problem, we have included two concepts, namely, monetary and compactness, into the traditional sequential pattern mining algorithm of our proposed method.

(1) Monetary: In general, sequential patterns that occur frequently in the sequential database are used to find the significance of the user buying sequences. However, in the business perspective, there is always a need to consider the price of an item. The reasons behind that are (a) some patterns that are frequently occurring in sequential database are not providing much profit and (b) the purchasing behavior of the user will be changed based on the price of an item. For example, items such as, shampoo, toothpaste, soap and hair oil are frequently bought by customers, but though expensive goods like gold and diamond are not frequently purchased, they provide better profit compared to frequently purchased items.

(2) Compactness: In most practical problems, particularly, pattern learning for managerial decision support, it is essential to push time constraint in the sequential pattern mining task. This is also observed with customer purchasing database that the buying behavior of the customers can be varied over time. Thus, there is a need to consider the time so that decision makers, who are trying to find the user sequence behavior, can develop better marketing and product strategies. The advantage of compactness is that, it allows mining sequential patterns that occur within a reasonable time span. Furthermore, if the time span for the buying sequence is too lengthy, by diminishing the importance of the patterns it permits the mining algorithm to provide better solutions for decision makers.

In order to mine more significant patterns, we push the concept of monetary and compactness to the sequential mining process along with the frequency to discover the CFM-patterns. The number of purchases made within a definite period, where a higher frequency specifies higher loyalty is called Frequency. Monetary is the quantity of money spent during a particular period, and a higher value reveals that the company should pay more attention to that customer. Compactness signifies that the number of purchases made by the customer must be within a reasonable time period. If the mining process includes the above three concepts, the decision makers can simply categorize their customers, and provide a specific score to their customers based on these concepts. In addition, the mined patterns can assist the company to find out which customers are more significant.

3.1. **CFM-PrefixSpan algorithm.** In this section, we describe an efficient algorithm, CFM-PrefixSpan, for mining all the CFM- patterns from sequence databases. The CFM-PrefixSpan algorithm is developed by modifying the eminent PrefixSpan algorithm, which uses the pattern growth methodology for mining the frequent sequential patterns recursively. At first, we define Subsequence, Compact subsequence, Compact Frequent subsequence, Monetary subsequence and Compact Frequent Monetary subsequence because the proposed CFM-PrefixSpan algorithm utilizes these definitions. Then, we provide a brief description about the proposed CFM-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 signifies the time at which p_j occurs, $1 \leq j \leq n$ and $t_{j-1} \leq t_j$ for $2 \leq j \leq n$. P denotes a set of items in the database D.

Definition 3.1. (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 sub sequence of S only if, (a) itemset S_s is a subsequence of S, $S_s \in S$, (b) $t_1 < t_2 < \cdots < t_m$ where, t_1 is the time at which q_1 occurred in S_s, $1 \leq r \leq m$.*

Definition 3.2. (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 C_T be the predefined compact threshold. S_s is known to be a compact subsequence of S if and only if (a) S_s is a subsequence of S, and (b) the compactness constraint is satisfied, i.e., $t_m - t_1 \leq C_T$.*

Definition 3.3. (Compact Frequent subsequence): *Let D be a sequential database containing itemsets, I and C_T be the predefined compact threshold. S_s is said to be a compact frequent subsequence of D if and only if (a) S_s is a subsequence of D, (b) the compactness constraint is satisfied, i.e., $t_m - t_1 \leq C_T$, and (c) S_s is a frequent subsequence of database, D.*

Definition 3.4. (Monetary 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_m be the predefined monetary threshold. S_s is said to be the monetary subsequence of S if and only if (a) S_s is a subsequence of S, and (b) the monetary constraint is satisfied, i.e., $\left(\frac{M_1+M_2+\cdots+M_m}{m}\right) \geq T_m$.*

Definition 3.5. (Compact Frequent Monetary subsequence): *Let D be a sequential database containing itemsets (I). C_T 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, (a) S_s is a subsequence of D, (b) the compactness constraint is satisfied, i.e., $t_m - t_1 \leq C_T$, (c) S_s is a frequent subsequence of database, D and (d) the monetary constraint is satisfied, i.e., $\left(\frac{M_1+M_2+\cdots+M_m}{m}\right) \geq T_m$.*

The following provides a detailed explanation of the important steps involved in the proposed CFM-PrefixSpan algorithm. The CFM-PrefixSpan algorithm is outlined as follows.

Input: A sequence database D, and the minimum support threshold min_sup, monetary table M_T, Predefined compact threshold C_T, Predefined monetary threshold T_m.
Output: The complete set of CFM-sequential patterns β.
Method: Call *CFM_PrefixSpan* $(\langle\ \rangle, 0, D, M_T)$.
Subroutine: *CFM_PrefixSpan* $(\alpha, l, D|_\alpha, M_T)$
Parameters: α is sequential pattern; l is the length of α; $D|_\alpha$ is the α-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, find the set of compact frequent items f such that
    a) f can be assembled to the last element of α to form a sequential pattern; or
    b) $\langle f\rangle$ can be appended to α to form a sequential pattern.

2. For each compact frequent item f, append it to α to form a sequential pattern α'.
3. For each α',

   a) check monetary using M_T;

4. Create a set β from α' by substituting the findings of Step 3.
5. For each α', construct α'-projected database $D|_{\alpha'}$, and call *PrefixSpan* $(\alpha', l + 1, D|_{\alpha'}, M_T)$.

Step 1: Finding 1-CFM patterns

At first, the sequential database D and monetary table M_T are given to the proposed CFM-PrefixSpan algorithm. We mine the 1-CFM sequential patterns from the sequential database by scanning the database once. The 1-CF patterns (compact frequent) which satisfy the predefined compact threshold and support threshold are mined from the sequential database by simply scanning the database. Then, we apply the monetary constraint on the 1-CF patterns so that we can obtain a set of 1-CFM patterns.

Example 3.1. *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* $(C_T = 4)$ *and predefined support* $(\text{min\_sup} = 2)$ *as follows* $[(\text{a} \rightarrow 2), (\text{b} \rightarrow 3), (\text{c} \rightarrow 3), (\text{d} \rightarrow 2), (\text{f} \rightarrow 1))]$. *In this set, the patterns that satisfy the compact threshold and support threshold known as 1-CF patterns are as follows* $[(\text{a} \rightarrow 2), (\text{b} \rightarrow 3), (\text{c} \rightarrow 3), (\text{d} \rightarrow 2)]$. *Next, we compute the monetary of the 1-CF patterns obtained from the previous step,* $[\text{a} \rightarrow (2,5), \text{b} \rightarrow (3,10), \text{c} \rightarrow (3,15), \text{d} \rightarrow (2,2)]$. *Based on the monetary threshold* $(T_m = 10)$*, we obtain the following set of 1-CFM patterns as follows* $\{\text{b } and \text{ c}\}$.

Table 1. Sequential database

Customer ID	Sequence
1	< (a, 1), (b, 3), (c, 4), (d, 4), (f, 5) >
2	< (b, 1), (c, 2), (d, 3) >
3	< (a, 3), (b, 4), (c, 4) >

Table 2. Monetary table

Item	Monetary value
a	5
b	10
c	15
d	2
f	10

Step 2: Dividing search space

The mined 1-CF patterns are then used to construct the projected database that 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 we can obtain k disjoint subsets from the sequential database using the complete set of 1-length compact frequent patterns.

Example 3.2. *Here, we form the projected database for the 1-CF patterns such as* $\{\text{a}, \text{b}, \text{c}$ *and* $\text{d}\}$. *The steps used for constructing the projected database of the pattern* $< \text{a} >$ *are as follows: By looking at the first sequence in the database,* $< \text{a} >$ *has a time stamp value*

Table 3. Projected database for 1-length compact frequent pattern

< a >	< (b, 3), (c, 4), (d, 4), (f, 5) > < (b, 4), (c, 4) >
< b >	< (c, 4), (d, 4), (f, 5) > < (c, 2), (d, 3)> < (c, 4) >
< c >	< (d, 4), (f, 5) > < (d, 3) >
< d >	< (f, 5) >

of 1. So, the projection based on the first sequence is obtained by taking the postfixes of pattern $<$ a $>$ *(sequences after the time stamp 1) in the first sequence. In a similar way, we obtain the projection for the rest of sequences present in the sequential database. The projected database for the pattern* $<$ a $>$ *contains* $< \mathrm{(b,3),(c,4),(d,4),(f,5)} >$ *and* $< \mathrm{(b,4),(c,4)} >$*. Similarly, the projection is done for other 1-CF patterns. Table 3 shows the projected database of all one length CF patterns in the projection set.*

Step 3: Finding subsets of sequential patterns

In this step, we mine a set of 2-length compact frequent patterns by scanning the projected database once. Then, we obtain the set of 2-CFM patterns by applying the monetary constraint on the 2-length compact frequent patterns. Again, the projected database is formed with the help of mined 2-CF patterns and this process is repeated recursively until all CFM patterns are mined.

Example 3.3. *The projected database formed by the 1-length CF sequential patterns is utilized for mining all 2-length CFM sequential patterns. The procedure employed for mining 2-length CFM sequential patterns having prefix* $<$ a $>$ *is as follows: By scanning the projected database once, we obtain the count of the compact frequent items which is represented as,* $[(\mathrm{a} \rightarrow 0), (\mathrm{b} \rightarrow 2), (\mathrm{c} \rightarrow 1), (\mathrm{d} \rightarrow 0), (\mathrm{f} \rightarrow 0)]$*. In this set, the patterns that are satisfying the compact threshold and support threshold are given as,* $[(\mathrm{b} \rightarrow 2)]$*. The mined 2-CF sequential pattern is* $\{ab\}$*. Then, we apply the monetary constraint on the mined 2-CF sequential pattern so that, we can obtain the 2-CFM pattern* $[< \mathrm{ab} > \rightarrow (2, 7.5)]$*. There is no 2-length CFM sequential pattern having a prefix* $<$ a $>$ *since the pattern* $<$ ab $>$ *has not satisfied the given monetary threshold. Again, we form the 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 CFM patterns with prefix* $<$ a $>$ *recursively. The aforementioned procedure is repeated for other 1-CF patterns* $<$ b $>$*,* $<$ c $>$ *and* $<$ d $>$*. The mined CFM-patterns are* $\{< \mathrm{b} >, < \mathrm{bc} >, < \mathrm{c} >\}$*.*

4. **Results and Discussion.** The experimental results of the proposed CFM-PrefixSpan algorithm for effectual mining of CFM patterns is described in this section and the comparative analysis of the proposed algorithm is also presented for various performance measurements.

4.1. **Experimental set up and dataset description.** The proposed CFM-PrefixSpan algorithm is programmed using JAVA (jdk 1.6) through the help of Netbeans IDE. The experimental set up includes the PC running on core i3 processor with 4 GB RAM. The performance of the proposed algorithm has been evaluated using the synthetic datasets as well as real life datasets. ***Synthetic dataset:*** Here, we have generated a sequential database that contains 10,000 sequences of 10 items. ***Real life datasets:*** We make use of "MSNBC.com Anonymous Web Data" [7] that was taken form UCI machine learning repository. This data describes the page visits of users who visited msnbc.com on September 28, 1999. Visits are recorded at the level of URL category ("frontpage", "news", "tech", "local", "opinion", "on-air", "misc", "weather", "health", "living", "business", "sports", "summary", "bbs" (bulletin board service), "travel", "msn-news", and "msn-sports") and are recorded in time order.

4.2. **Experimentation.** The sample database taken for experimentation is given in Table 1 and the monetary table is given in Table 2. Then, we fed the database and monetary table as an input to the proposed CFM-PrefixSpan algorithm for effectual mining of CFM sequential patterns. Initially, we mined the 1-CFM patterns based on the thresholds, $C_T = 4$, $\min\_\text{sup} = 2$, $T_m = 10$. Subsequently, the projection was done based on the mined 1-length compact frequent patterns. The projected database for the 1-CF pattern is presented in the Table 3. Finally, we obtained a complete set of CFM patterns for the given input sequential database. The obtained complete set of CFM pattern set is $\{<\text{b}>, <\text{bc}>, <\text{c}>\}$.

The comparative results of the PrefixSpan with our proposed CFM-PrefixSpan algorithm are given in Table 4. It clearly ensures that the proposed algorithm provides lesser number of sequential patterns compared to PrefixSpan algorithm. The PrefixSpan algorithm contains all the less profitable and longer time length sequential patterns (< a >, < ab >, < ac >, < d > and < bd >) but the proposed algorithm generates CFM sequential patterns that contain only the profitable and valuable sequential patterns (< b >, < bc > and < c >). So from a business perspective, CFM-prefixSpan algorithm is more suitable for developing better business strategies compared to PrefixSpan algorithm.

Table 4. Comparison of the proposed algorithm with PrefixSpan algorithm

	CFM-sequential Pattern	**Sequential Pattern**
< a >		< a >, < ab >, < ac >
< b >	< b >, < bc >	< b >, < bc >, < bd >
< c >	< c >	< c >
< d >		< d >

4.3. **Performance measurement.** The performance of the proposed algorithm in mining of sequential patterns is analyzed with the help of three different experiments: ***a) the effect of compact threshold, b) the effect of support value and c) scalability analysis***. In the first set of experiment, the *compact threshold* is varied to different value and observed for the total number of sequence generated and the computation time. The results are compared for both the naive algorithm as well as proposed algorithm. In the second set of experiment, support is varied significantly and measured for the total number of sequence generated, computation time and memory usage to find the performance of these algorithms. Similar way, the scalability analysis is also carried out by varying the size of the database.

4.4. **Comparative analysis.**

1) Effect of compact threshold:

For performance comparison, the CFM-PrefixSpan and the PrefixSpan algorithms are applied on the synthetic datasets to discover a set of sequential patterns. These two algorithms are compared in terms of number of significant sequential patterns obtained

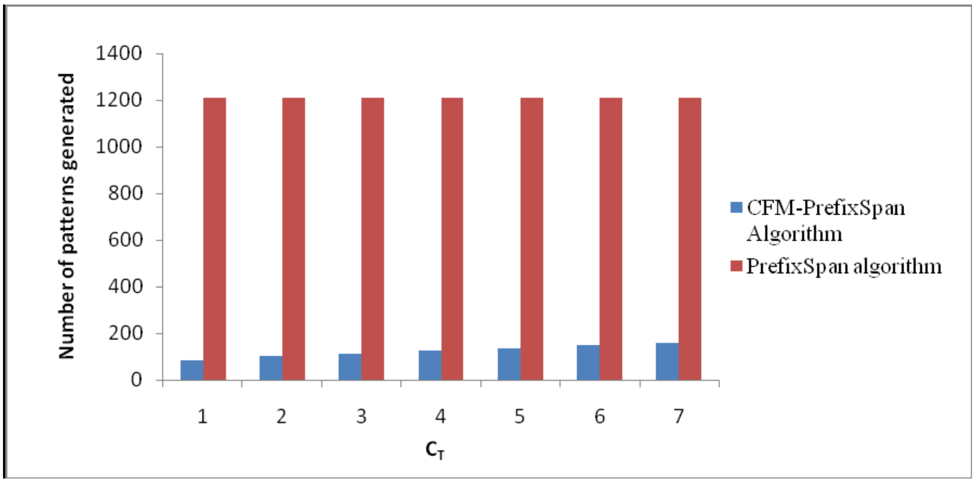

FIGURE 1. Comparison graph of the min_sup = 1000 and $T_m = 10$

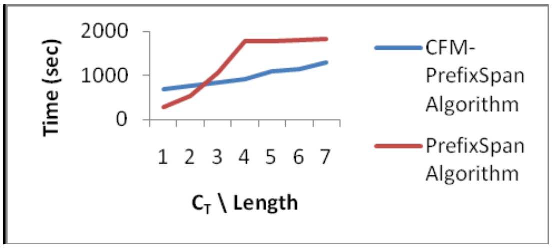

FIGURE 2. Run time performance of the algorithm

for different support thresholds. By inputting the min_sup = 1000 and $T_m = 10$, the results are computed by varying the C_T and the obtained results are given in the Figure 1. From the graphs shown in Figure 1, it is obvious that the number of the sequential patterns obtained by the proposed algorithm is reduced significantly as compared with the PrefixSpan algorithm.

Then, we take the computation time, which is one of the important parameter to find the complexity of the algorithm. For the constant value of min_sup = 1000 and $T_m = 10$, we have discovered a set of sequential patterns and analyzed that the time taken by the algorithms for various threshold C_T (for CFM-PrefixSpan) and length (PrefixSpan). The time required to complete the mining task is computed and the values are plotted in a graph shown in Figure 2. By comparing the computational complexity, the proposed algorithm takes less computation time than the PrefixSpan algorithm for higher threshold values.

2) Effect of support values:

In order to analyze the effects of the algorithms in terms of support value, the synthetic and real datasets are given to the PrefixSpan and CFM-Prefixspan algorithm. These algorithms are compared for the number of sequences obtained, computation time and the memory usage. The values obtained through the experimentation are plotted as graphs that are shown in Figures 3-8. From the graphs, we can understand that the number of sequence generated from the CFM-Prefixspan algorithm is less compared with Prefixspan algorithm. This signifies that the most important rules are only mined by the CFM-Prefixspan algorithm. When analyzing the run time performance, the proposed algorithm outperformed the previous algorithm in both synthetic and real dataset. For synthetic datasets, the proposed algorithm achieved six times more efficiency in computation time and for real datasets also, the proposed algorithm achieved comparable results.

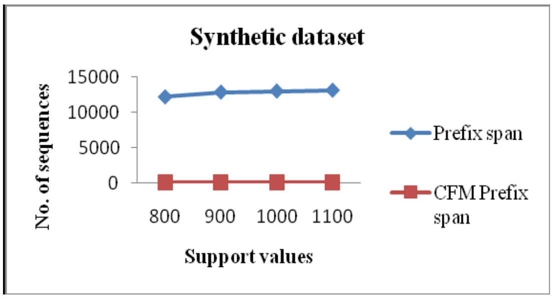

FIGURE 3. Number of sequences generated for synthetic dataset

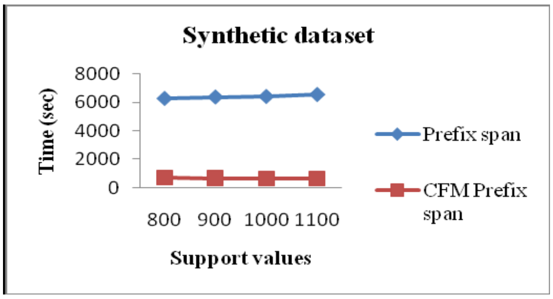

FIGURE 4. Run time performance of the algorithm for synthetic dataset

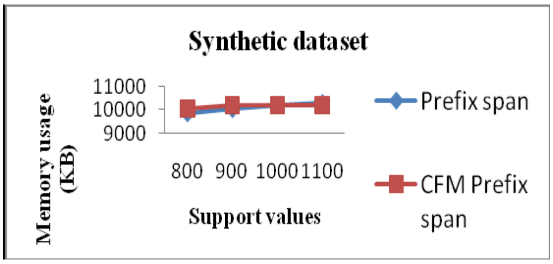

FIGURE 5. Memory usage of the algorithms in synthetic dataset

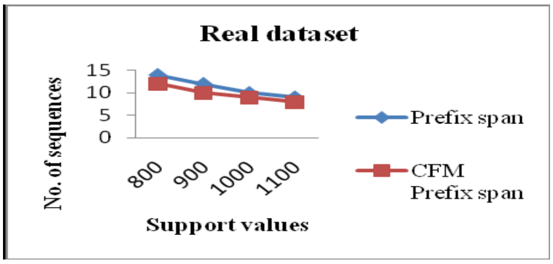

FIGURE 6. Number of sequences generated for real dataset

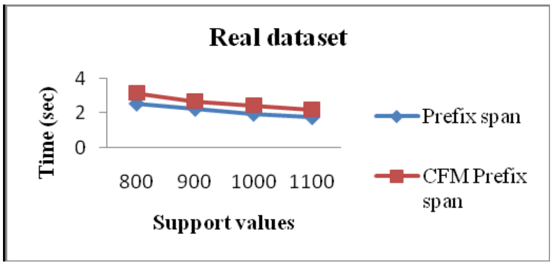

FIGURE 7. Run time performance of the algorithm for real dataset

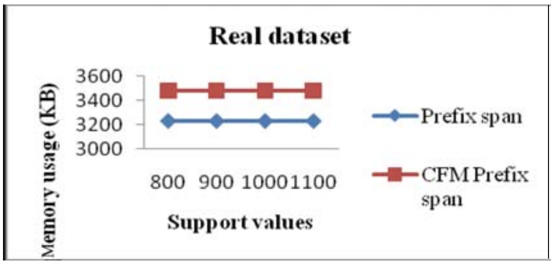

FIGURE 8. Memory usage of the algorithms in real dataset

3) Effect of scalability:

The scalability of the algorithms is analyzed in both synthetic and real datasets using number of sequences, computation time and memory usage. For various numbers of records, the number of sequences obtained, computation time and memory usage are calculated and the values obtained through the experimentation are plotted as graphs that are shown in Figures 9-14. When comparing the sequential patterns generated, the proposed algorithm mines the more significant patterns in synthetic and real datasets. Furthermore, the computation time required to mine the sequential pattern is comparable with the PrefixSpan algorithm. Comparing the computation time for synthetic datasets, the proposed algorithm needs very less time compared with the naïve algorithm. Similarly, the performance of the proposed algorithm is also effective in comparing the memory usage.

5. **Conclusion.** We have presented an efficient algorithm, CFM-PrefixSpan algorithm, for mining all CFM sequential patterns from the customer transaction database. The

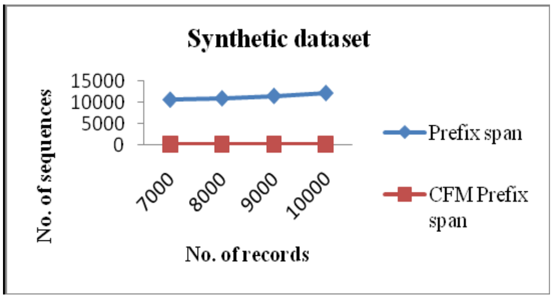

Figure 9. Number of sequences generated for synthetic dataset

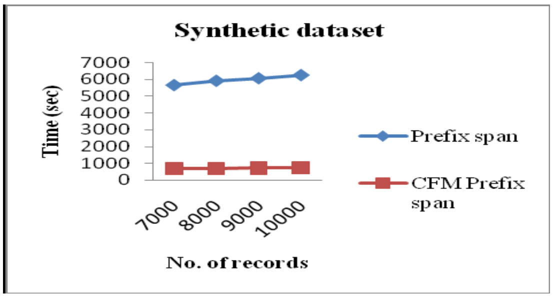

Figure 10. Run time performance of the algorithm for synthetic dataset

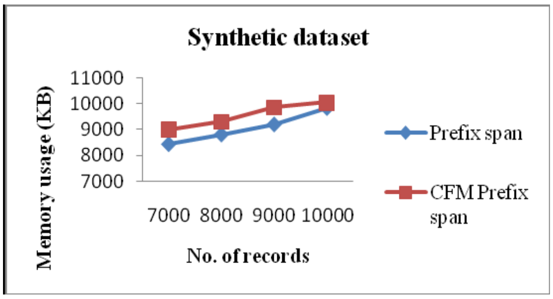

Figure 11. Memory usage of the algorithms in synthetic dataset

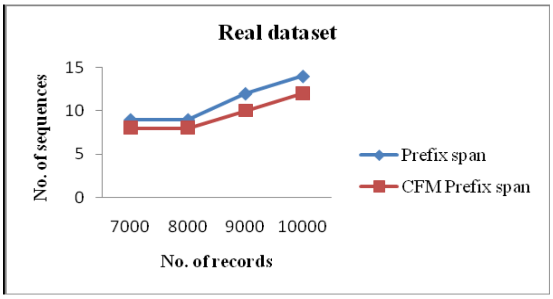

Figure 12. Number of sequences generated for real dataset

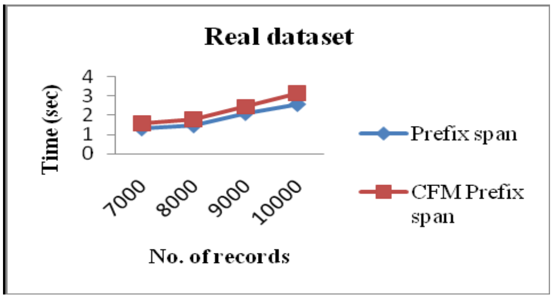

Figure 13. Run time performance of the algorithm for real dataset

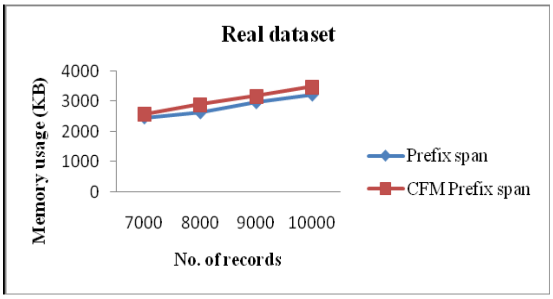

Figure 14. Memory usage of the algorithms in real dataset

CFM-PrefixSpan algorithm employed a pattern-growth methodology that finds sequential patterns by utilizing a divide-and-conquer strategy. We have used two concepts namely, monetary and compactness that are derived from aggregate and duration constraints in addition to frequency for mining interesting and valuable sequential patterns. In our algorithm, the sequence database has been recursively projected into a set of smaller projected databases based on the compact frequent patterns. Besides, CF-sequential patterns have been discovered in each projected database by exploring only locally compact frequent items and then, the CFM sequential patterns are discovered. The discovered CFM sequential patterns signify valuable information on customer purchasing behavior and ensure that all patterns have reasonable time spans with good profit. The experimental results have showed that the effectiveness of sequential pattern mining algorithms can be improved significantly by incorporating monetary and compactness into the mining process.

Acknowledgements. The research work has been done at Thapar University, Patiala as a research scholar by the corresponding author. The authors gratefully acknowledge the support provided by the Thapar University. They express their gratitude to the Editor and anonymous reviewers for their valuable comments and constructive suggestions.

REFERENCES

[1] M.-Y. Lin and S.-Y. Lee, Efficient mining of sequential patterns with time constraints by delimited pattern growth, *Knowledge and Information Systems*, vol.7, no.4, pp.499-514, 2005.

[2] C. Fiot, A. Laurent and M. Teisseire, Extended time constraints for sequence mining, *Proc. of the 14th International Symposium on Temporal Representation and Reasoning (TIME'07)*, Alicante, Spain, pp.105-116, 2007.

[3] J. Bisaria, N. Srivastava and K. R. Pardasani, A rough set model for sequential pattern mining with constraints, *Proc. of the International Journal of Computer and Network Security*, vol.1, no.2, 2009.

[4] E. Chen, H. Cao, Q. Li and T. Qian, Efficient strategies for tough aggregate constraint-based sequential pattern mining, *Information Sciences*, vol.178, no.6, pp.1498-1518, 2008.

[5] F. Masseglia, P. Poncelet and M. Teisseire, Efficient mining of sequential patterns with time constraints: Reducing the combinations, *Expert Systems with Applications*, vol.36, no.2, pp.2677-2690, 2009.

[6] J. Bisaria, N. Shrivastava and K. R. Pardasani, 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, 2009.

[7] *Anonymous Web Data*, Sethttp://archive.ics.uci.edu/ml/datasets/MSNBC.com+Anonymous+Web +Data.

[8] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal and M.-C. Hsu, Mining sequential patterns by pattern-growth: The PrefixSpan approach, *IEEE Transactions on Knowledge and Data Engineering*, vol.16, no.10, 2004.

[9] Q. Zhao and S. S. Bhowmick, Sequential pattern mining: A survey, *Technical Report, CAIS*, Nanyang Technological University, Singapore, 2003.

[10] J. Pei, J. Han and W. Wang, Constraint-based sequential pattern mining: The pattern-growth methods, *Journal of Intelligent Information Systems*, vol.28, no.2, pp.133-160, 2007.

[11] S. Hou and X. Zhang, Alarms association rules based on sequential pattern mining algorithm, *Proc. of the 5th International Conference on Fuzzy Systems and Knowledge Discovery*, vol.2, pp.556-560, Shandong, China, 2008.

[12] T. Sobh, *Innovations and Advanced Techniques in Computer and Information Sciences*, Springer, 2007.

[13] F. Masseglia, P. Poncelet and M. Teisseire, Incremental mining of sequential patterns in large databases, *Data & Knowledge Engineering*, vol.46, no.1, pp.97-121, 2003.

[14] J. D. Parmar and S. Garg, Modified web access pattern (mWAP) approach for sequential pattern mining, *Journal of Computer Science*, vol.6, no.2, pp.46-54, 2007.

[15] S. Orlando, R. Perego and C. Silvestri, A new algorithm for gap constrained sequence mining, *Proc. of the ACM Symposium on Applied Computing*, Nicosia, Cyprus, pp.540-547, 2004.

[16] R. Agrawal and R. Srikant, Mining sequential patterns, *Proc. of the 11th International Conference on Data Engineering*, Taipei, Taiwan, pp.3-14, 1995.

[17] S. Myra, Web usage mining for Web site evaluation, *Communications of the ACM*, vol.43, no.8, pp.127-134, 2000.

[18] R. Srikant and R. Agrawal, Mining sequential patterns: Generalizations and performance improvements, *Proc. of the 5th International Conference on Extending Database Technology (EDBT'96)*, Avignon, France, pp.3-17, 1996.

[19] C. Antunes and A. L. Oliveira, Sequential pattern mining with approximated constraints, *Proc. of the International Conference on Applied Computing*, pp.131-138, 2004.

[20] Y.-H. Hu, *The Research of Customer Purchase Behavior Using Constraint-Based Sequential Pattern Mining Approach*, Ph.D. Thesis, 2007.