Applied Constraints on Sequential Pattern Mining with Prefixspan Algorithm

Kalpesh A. Kshatriya[#1], Bhavikkumar M. Patel[*2], Hitesh B. Patel[@3]

[#]Lecturer,I.T. DEPARTMENT BBIT, Vallabh Vidhyanagar, Anand, India
[*]Lecturer,COMPUTER ENGINEERING DEPARTMENT BBIT, Vallabh Vidhyanagar, Anand, India
[@]Lecturer,I.T. DEPARTMENT BBIT, Vallabh Vidhyanagar, Anand, India

Abstract - *Sequential pattern mining is the process of applying data mining techniques to a sequence database for the purpose of discovering the correlation that exist among an order list of events. Here the PrefixSpan [1] sequential pattern mining algorithm is used to generate sequential patterns from the dataset. After applying any sequential pattern mining algorithm on large dataset there could be a huge number of sequential patterns generated which are very hard to understand and hard to use by the users[2]. Users are often interested in only small subset of such patterns so by inserting several constraints with the sequential pattern mining algorithm we can restrict the algorithm from generating such a huge number of patterns. Here we study constraints like Item, Duration and Length of Transaction with the PrefixSpan algorithm in order to handle the large database. Less number of sequential patterns is generated when we use the PrefixSpan algorithm with Item, Duration and Length of Transaction constraint.*

I. Introduction

As a young research field, data mining has made significant progress and covered a broad spectrum of applications since 1980. Today, data mining is used in a vast array of areas. It is no surprise that data mining, as a truly interdisciplinary subject, can be defined in many different ways. Even the term data mining does not really present all the major components in the picture. To refer to the mining gold from rocks or sand, we say gold mining instead of rock or sand mining. Analogously, data mining should have been more appropriately named "knowledge mining from data," which is unfortunately somewhat long. However, the shorter term, knowledge mining may not reflect the emphasis on mining from large amounts of data. Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious nuggets from a great deal of row material. Thus, such misnomer carrying both "data" and "mining" became a popular choice. In addition many other terms have a similar meaning to data mining – for example, knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, data dredging [3].

One of the active research areas is sequential pattern mining which finds interesting sequential patterns among the large database. It finds out frequent subsequences as patterns from a sequence database. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining sequential patterns from their database [4,5].

II. Introduction to Sequential Pattern Mining

Sequential pattern mining is the mining of frequently occurring ordered events or subsequence's as patterns. An example of a sequential pattern is "Customers who buy a Samsung digital camera are likely to buy an HP colour printer within a month." For retail data, sequential patterns are useful for shelf placement and promotions. This industry, and other businesses, may also use sequential patterns for targeted marketing, customer retention, and many other tasks. Other areas in which sequential patterns can be applied include Web access pattern analysis, weather prediction, and network intrusion detection. The sequential pattern mining problem was first introduced by Agrawal and Srikant in 1995 based on their study of customer purchase sequences, as follows: Given a set of sequences, where each sequence consists of a list of events (or elements) and each event consists of a set of items, and given a user-specified minimum support threshold of min sup, sequential pattern mining finds all frequent subsequence's, that is, the subsequence's whose occurrence frequency in the set of sequences is no less than min support [4].

An itemset is a nonempty set of items A sequence is an ordered list of events. A sequence *s* is denoted $<e_1e_2e_3....e_l>$, where event e_1 occurs before e_2, which occurs before e_3, and so on. Event e_j is also called an element of *s*. In the case of customer purchase data, an event refers to a shopping trip in which a customer bought items at a certain store. The event is thus an itemset, that is, an unordered list of items that the customer purchased during the trip. The itemset (or event) is denoted $(x_1x_2....x_q)$, where x_k is an item. For brevity, the brackets are omitted if an element has only one item, that is, element (x) is written as x. Suppose that a customer made several shopping trips to the store. These ordered events form a sequence for the customer. That is, the customer first bought the items in s_1, and then later bought the items in s_2, and so on. An item can occur at most once in an event of a sequence, but can occur multiple times in different events of a sequence. The number of instances of items in a sequence is called the length of the sequence.

III. Problem Formulation

This section presents the formal definition of the problem of sequential pattern mining Let D be a database of customer transactions, $I=\{I_1,I_2,....,I_n\}$ be a set of m distinct attributes called items. A Sequence is an ordered list of itemsets. A sequence s is denoted by $< s_1\ s_2\ ......s_l >$, where s_i is an itemsets, i.e. $s_i \subseteq I$ for $1\leq i \leq l$. s_i is also called an element of the sequence. Since an element is a set, and the order of its items is not important. Length of a sequence is the number of instance of items in that sequence. A sequence with length k is called a k sequence [4].

A sequence $\alpha= < a_1\ a_2\ ....a_n>$ is called subsequence of the $\beta= <b_1\ b_2\ ...b_m>$, and denoted as $\alpha \subseteq \beta$, if there exist integers $1\leq j_1<j_2<...<j_n\leq m$ such that $a_1 \subseteq bj_1$, $a_2 \subseteq bj_2$, ... , $a_n \subseteq bj_n$. Sequence α is also called the super sequence of β. A sequence α is called a frequent sequence pattern in sequence database S, if the number of tuples in S that contain α is greater than or equal to a given positive integer γ , called support threshold , or minimum support , i.e., Support $(\alpha) \geq \gamma$. A sequence α is called a frequent Max sequential pattern in sequence database S, if α is a frequent sequential pattern in S, and there exists no frequent sequential pattern β in S, such that β is a proper super sequence of α [4].

The problem of sequential pattern mining is to find the complete set of frequent sequential patterns in a sequence database, for a given minimum support threshold[4].

A. Motivation

Sequential pattern mining, which extracts frequent subsequence's from a sequence database, has attracted a great deal of interest during the recent data mining research because it is the basis of many applications such as customer purchasing behaviour, web user analysis, stock trend prediction, DNA sequence analysis and using the history of symptoms to predict certain kind of disease. University and college Libraries stores the student book issue records electronically we can do the sequential pattern mining on such kind of records. By analysing sequential patterns on library transaction records we can improve the library facility for example library can provide the recommendation facility to the readers and also manage the books layout [6].

B. Types of Sequential Pattern Mining algorithms

There are broadly two types of sequential pattern mining algorithms available Apriori based and pattern growth based. In Apriori based approaches like AprioriAll [4] and GSP [7] large number of candidate sets are generated so multiple scanning are required to prune the candidate set. On the other hand in pattern-growth based approach like FreeSpan [8] and PrefixSpan [1] there is no candidate generation process they work on the projected database approach which saves a lot of computation cost. PrefixSpan algorithm is used to generate the frequent sequences from the transaction data. After applying sequential pattern mining on large database there could be a large number of sequential patterns generated so constrained based sequential mining is needed which extracts the sequential patterns which are of users interest [10]. For effectiveness and efficiency considerations, constraints are essential in many sequential pattern mining applications.

C. Scope

With Constrained based PrefixSpan algorithm we can generate the sequential patterns from transaction data in efficient and effective manner.

D. Available Sequential pattern Mining Approaches.

Generally two types of approaches are there for sequential pattern mining [9, 11, 15].

- Apriori Based
- Pattern-Growth Based

1) Apriori Based Approache

Apriori-based Sequential Pattern mining was first proposed in [4] termed AprioriAll algorithm set the ground for engender the sequential pattern mining algorithms that are largely depends on the apriori property [12] and uses the Apriori-generate join scheme to produce candidate sequences. The apriori property says that "any sub-pattern of a frequent pattern must be frequent" [12]. It is also delineate as antimonotonic, in that if the sequence cannot gratify the minimum support value, all of its super sequences will also let down the test. Some of the Aprioir-based sequential pattern mining algorithms are AprioriAll [4] and GSP [7].

1.1 AprioriAll Algorithm

This Sequential pattern mining was first proposed in [4]. If we have the library transaction database with three attributes student-id, transaction time i.e. the book issue date and book-id, the problem of mining sequential pattern is to find the maximal sequences from all frequent sequences that satisfy the minimum support value. There are five steps in the AprioriAll algorithm. Identify

Sort Phase: In this phase database is sorted with student-id as the major key and transaction time as the minor key.And after that the database is converted into the sequence database.

Litemset Phase: In this phase the database is scanned to obtain frequent 1-sequences and also large 1-itemset. The set of Litemset is then mapped to a set of contiguous integers.

Transformation Phase: In this phase the sequence database is transformed into a set of sequence database by those large itemsets. Now if the sequence database does not contain any large itemset, then that sequence is not carry in the transformed sequence.

Sequence Phase: In this phase multipasses are performed over the database to found the sequential pattern. This phase is somewhat similar to the candidate generation and test step of the apriori algorithm [12]. In the AprioirAll algorithm we first found the frequent 1-sequences from the transformed database and this frequent 1-sequences becomes the seed set for finding the frequent 2-sequences and the process will carryout upto the point when no more frequent sequence generation are possible.

Maximal phase: After the sequence phase we see that lots of frequent sequences are generated so to reduce this huge number of frequent sequences we are only interested in the maximal sequences [4,10]. The sequences are called maximal sequences if s is not contained in any other sequences.

One of the drawbacks of AprioriAll algorithm was that it generates lots of candidate sequences so it is very time consuming process to prune the candidate sequences [3, 1].

2) Pattern-GrowthBased Algorithm

Promptly after the Apriori-Based approaches in the mid-1990s, the pattern growth approaches appear in the decade of 2000 onwards as a solution to the issue of generate-and-test [8,1]. The crucial feeling is to avert the candidate generation step, and to target the search on restricted portion of the original database. The search space partitioning factor shows an essential role in pattern-growth. Approximately every pattern-growth algorithm starts by making a representation of the database to be mined, then suggests a way to partition the search space, and generates as less candidate sequences as possible by expanding on the previously mined frequent sequences. FreeSpan [8] and PrefixSpan [1] are the two algorithms started using projected database being the most influencing.

3) PrefixSpan Algorithm

This algorithm was proposed in [1] with the aim of reducing the huge number of candidate sequences generated in the Apriori-Based algorithms [1, 11]. This algorithm examines only the prefix subsequence's and projects only their corresponding postfix subsequence's into projected database.

The procedure of PrefixSpan algorithm is described as follows [1].

Step1: Find length one sequential patterns

At first the PrefixSpan algorithm finds all frequent items which are 1-length sequential patterns by scanning the sequential database with respect to the given minimum support these frequent one length sequences used as a prefix for generating the projected database.

Step2: Divide search space

Divide search space into prefixes whose support is greater than the minimum support.

Step3: Find subset of sequential patterns

The subset of sequential patterns can be searched by constructing projected database and mine each recursively.

4) Working of PrefixSpan Algorithm with Example

Table 1 shows the sample library dataset where first column represent the transaction date on which the student issued books, second column represent student-id and the last column represent different books issued by the students at particular date; here books are represented by their book id.

Now as the pre-processing phase the transaction database of Table1 is first converted into the sequence database as shown in Table 2. In Table 2 first column represent student-id and the second column represent different books issued by students at different date. Here the bracket shows that the particular books are issued together. For simplification we use the different characters instead of book-id like 1090 is denoted by A ,1094 is denoted by B, 2050 is denoted by C and 3021 is denoted by D .

First step of PrefixSpan algorithm is to generate the frequent 1-length sequences; here we set the value of minimum support as 2 means the patterns are frequent only if they come in 2 or more than 2 sequences in the sequence database.

Here frequent 1-length sequences for the sample database of Table2 are <A>:4, <B>:3, <C>:2, <D>:2

Table 1 Sample Transaction Dataset of Library

Transaction Date	Student ID	Book issued
21/08/2012	11CP606	1090,1094
22/08/2012	11CP607	1090
10/09/2012	11CP607	1090,3021
16/08/2012	11CP615	1090,1094
30/08/2012	11CP615	1090,1094
04/09/2012	11CP615	2050,3021
07/09/2012	11CP615	2050,1094
16/08/2012	11CP618	1090,1094
18/08/2012	11CP618	1094
21/08/2012	11CP618	1090
23/08/2012	11CP618	2050
31/08/2012	11CP618	1090

Table 2 Sequence Database

11CP606	<(AB)>
11CP607	<A(A,D)>
11CP615	<(AB)(AB)(CD)(BC)>
11CP618	<(AB)BACA>

After generating the frequent 1-length sequences these 1-length sequences are used to divide the search space and generates the projected database based on these 1-length sequences [1]. Table 3 shows the projected database for every 1-length sequences. For prefix A and sequence <(AB)> subsequence for mining sequential pattern having prefix <A>, is <(_B)> and if the sequence is like <A(A,D)> so the subsequence is <(AD)>. So complete set of projected database for prefix <A> is <(_B)>, <(A,D)>, <(_B)(AB)(CD)(BC)>, <(_B)BACA>. Similarly we can generate the projected database for every frequent 1-length sequences. Table 3 shows the projected database for every 1-length sequences.

After frequent 1-length sequences frequent 2-length sequences are generated from the projected database, consider for example we have to generate the 2-length sequential patterns from <A> for this we have to scan the projected database of <A> and generate the 1-length pattern from this projected database.

Table 3 Projected Database for 1-length Frequent Sequences

Prefix	Projected (Postfix) Database
<A>	<(_B)>, <(A,D)>, <(_B)(AB)(CD)(BC)>, <(_B)BACA>
< B >	<(AB)(CD)(BC)> , <BACA>
< C>	<(_D)(CB)>
< D>	<(CB)>

Now we only consider those frequent 1-length patterns which satisfy the minimum support value and combine the prefix A with these 1-length patterns to generate the 2-length sequential pattern and after that recursively mine the other projected database for generating the three or more than three length patterns. The complete step of generating the sequential patterns with prefix <A> is shown in Table 4 here we see that for prefix <A> the generated patterns are <AA>, <AB>, <AC>, <AD>, <(AB)>, <AAC>, <ABD>, <(AB)A>, <(AB)B>, <(AB)C>, <(AB)AC>,<(AB)BC>. Similarly algorithm generates the sequential patterns for prefix <B>, <C> and <D>. Table 5 shows the total sequential patterns generated.

5. Handling Constraints in PrefixSpan Algorithm

Sequential pattern mining faces many challenges in both effectiveness and efficiency. On one hand there could be a large number of sequential patterns in large database [13, 16]. A user is often interested in only a small subset of such patterns. Display all sequential patterns may make the mining result hard to understand and hard to use. This leads in the effectiveness concern: can we only found the sequential patterns that are highly interested to users.

On the other hand mining a large number of sequential patterns from large database is naturally a computationally expensive task. If we target on only those sequential patterns interesting to users, we may be able to save a lot of computation cost by those uninteresting patterns.

Constraint based sequential pattern mining extracts the sequential patterns which are of user's interest [13, 14]. For effectiveness and efficiency consideration, constraints are necessary in many sequential pattern mining applications [16].

The aim of sequential pattern mining with constraint is to found the entire set of sequential patterns satisfying a user specified constraint C.

Constraint based sequential pattern mining can be classified into following groups [17].

- **Post-Processing** (Filtering out sequential patterns that do not satisfy user specified constraints after the actual discovery process).
- **Candidate filtering** (Application of constraints to reduce the number of processed candidates).
- **Dataset Filtering** (Selecting only those transactions in the sequence database that can possibly support patterns that satisfy user-specified constraints).

There is no meaning of inserting the constraint after generating all the sequential patterns (i.e. after Post-Processing) because this does not improves the efficiency of the sequential pattern mining algorithm. So we can insert the constraints in the sequential pattern mining algorithm at the time of candidate processing or we can push the constraint on the sequence database (i.e. at Pre-processing stage in that we only allow the transaction in the sequence database that satisfy user specified constraint).

Table 4 Generation of Sequential Patterns for Prefix <A>

Prefix	Projected Database	Frequent 1-length Pattern	Frequent Sequential Patterns
<A>	<(_B)> <(A,D)> <(_B)(AB)(CD)(CB)> <(_B)BACA>	<A>:3 <B>:2 <C>:2 <D>:2 <(_B)>:3	<AA>:3 <AB>:2<AC>:2 <AD>:2<(AB)>:3
<AA>	<(_D)> <(_B)(CD)(CB)> <CA>	<A>:1 <B>:1 <C>:2 <D>:1 <(_B)>:1 <(_d)>:1	<AAC>:2
<AAC>	<(_D)(CB)> <A>	<A>:1 <B>:1 <C>:1<(_D)>:1	---
<AB>	<(CD)(CB)> <ACA>	<A>:1 <B>:1 <C>:2 <D>:1	<ABC>:2
<ABC>	<(_D)(CB)> <A>	<A>:1 <B>:1 <C>:1 <(_D)>:1	---
<AC>	<(_D)(CB)> <A>	<A>:1 <B>:1 <C>:1 <(_D)>:1	---
<AD>	<(CB)>	<B>:1 <C>:1	---
<(AB)>	<(AB)(CD)(CB)> <BACA>	<A>:2 <B>:2 <C>:2 <D>:1	<(AB)A>:2<(AB)B>:2 <(AB)C>:2
<(AB)A>	<(_B)(CD)(CB)> <CA>	<A>:1 <B>:1 <C>:2 <D>:1	<(AB)AC>
<(AB)AC>	<(_D)(CB)> <A>	<A>:1 <B>:1 <C>:1 <(_D)>:1	---
<(AB)B>	<(CD)(CB)> <ACA>	<A>:1 <B>:1 <C>:2 <D>:1	<(AB)BC>
<(AB)BC>	<(_D)(CB)> <A>	<A>:1 <B>:1 <C>:1 <(_D)>:1	---

Prefix	Sequential Patterns
<A>	<AA> <AB> <AC> <AD> <(AB)> <AAC> <ABD> <(AB)A> <(AB)B> <(AB)C> <(AB)AC> <(AB)BC>
<B>	<BA> <BB> <BC> <BAC> <BBC>
<C>	----
<D>	----

In both Candidates filtering and Dataset filtering approach the efficiency of the sequential pattern mining algorithm improves because the search space is greatly reduces and also less number of patterns are generated which are of user specific.

Traditional sequential pattern mining algorithms only consider frequency constraint but they do not return sequential patterns of user interest while constraint based sequential pattern mining considers frequency constraint as well as user specified constraints and return user specified patterns for effectiveness and efficiency improvement of the traditional sequential pattern mining algorithm.

Here we can study PrefixSpan algorithm with constraint. Here we study three constraints Item constraint, Duration constraint and the length of Transaction constraint.

From these three constraints Item constraint is come under the candidate filtering group while other two constraints are come under the Dataset filtering group. Descriptions of these constraints are given below with examples.

5.1 Item Constraint

This is the candidate filtering constraint which only generates those sequential patterns which starts with the user specified Item constraint. For the sequence database of Table 2 if user sets the Item Constraint as A then the algorithm only generate the sequential patterns that start with the item A. So with the Item constraint A following sequential patterns are generated as the output

Prefix	Sequential Patterns
<A>	<AA> <AB> <AC> <AD> <(AB)> <AAC> <ABD> <(AB)A> <(AB)B> <(ABC)> <(AB)AC> <(AB)BC>

Here we see that all the generated sequential patterns are start with the Item constraint that is set by the user. We are not getting sequential patterns that start with Items other than A. We push this Item constraint in the PrefixSpan algorithm at the time of candidate generation. After generating frequent 1-length sequences the algorithm only generate the projected database for the user specified Item constraint and all the sequential patterns are mined from this projected database recursively. Here as we know this Item constraint comes under the candidate filtering group so proposed algorithm not generate the projected database and candidate sets for Items other than the user specified Item constraint which ultimately improves the effectiveness and the efficiency of the PrefixSpan algorithm.

5.2 Duration Constraint

This constraint comes under the Dataset filtering group which insert at the pre-processing step of the PrefixSpan algorithm. This constraint works on the date field on which the transactions are done by the users here user inputs two threshold value starting date and ending date with the PrefixSpan algorithm. When this constraint is set by the user the transaction database is filter out and only those transactions which comes under the starting date and ending date are consider for generating the sequence database. For example consider the transaction database of Table 1 if user set the duration constraint as 16/08/2012 (start date) and 23/08/2012 (end date) then PrefixSpan algorithm only generates the sequence database for the transactions which comes under this two dates then the final sequence database with this duration constraint is shown below.

Sequence Database after Duration Constraint

11CP606	<(AB)>
11CP607	<A>
11CP615	<(AB)>
11CP618	<(AB)BAC>

Now the PrefixSpan algorithm generates the sequential pattern from this reduced sequence database which improves the efficiency and effectiveness of the PrefixSpan algorithm. By using this constraint users generate the sequential patterns from specific duration of their interest.

5.3 Length of Transaction Constraint

This constraint also comes under the Dataset filtering group. When this constraint is set to some user specified threshold then algorithm generates the sequence database for only those users whose number of transactions is more than this threshold value. For the transaction database of Table3.1 if the threshold value for this constraint is set to three then the sequence database generated is as follows.

Sequence database after Length of Transaction Constraint

11CP615	<(AB)(AB)(CD)(BC)>
11CP618	<(AB)BACA>

Compare to the sequence database of Table 2 sequences for users 11CP606 and 11CP607 are not included in the final sequence database because the number of transactions for these users are not satisfy the user specified Length of Transaction threshold which is set to three here. By this constraint we can reduce the search space which ultimately reduces the number of sequential patterns. By this constraint we generally get the long sequential patterns.

5.4 Working of the PrefixSpan algorithm with Constraints

Flow of Algorithm is as follows:

Step1: In the pre-processing step only those transaction are occur in the sequence database which satisfy the user defined Duration and Length of Transaction Threshold

Step2: In this step all the 1-length sequences are generated from the sequence database.

Step3: In this step projected database are generated from the 1-length sequences. Now algorithm mines the projected database recursively to generate the two or more length patterns. Here algorithm also checks the Item constraint. If the Item constraint is set by the user then projected database is generated only for that item and the sequential patterns are recursively mined from this projected database.

Conclusions

In this study sequential pattern mining based on PrefixSpan algorithm is evaluate in detail. With this investigation it is realized that PrefixSpan algorithm for sequential pattern mining was proposed with the objective of reducing the huge number of candidate sequences generated in the Apriori Based algorithms. In this study it is realized that constraint based sequential pattern mining extracts the sequential patterns which are of user's interest. It is also observed that for effectiveness and efficiency consideration constraints are necessary in many sequential pattern mining applications.

References

[1] Han, Jiawei, Jian Pei, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and M. C. Hsu. "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth." In proceedings of the 17th international conference on data engineering, Chicago pp. 215-224. 2001.

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

[3] Jiawei Han, Micheline kamber, and jian Pei "Data mining concepts and techniques" , The Morgan Kaufmann series in Data Management Systems. 2012.

[4] Agrawal R. and Srikant R., "Mining sequential patterns" , Proceedings of the Eleventh International Conference on Data Engineeering, vol., no., pp.3,14, 6-10 Mar 1995.

[5] Agrawal R., Imielinski T. And Swami A.N. , "Mining association rules between sets of items in large databases", In proceedings of the ACM SIGMOD International Conference on Managment of Data, Washington D.C., 207-216 1993.

[6] Sitanggang, Imas Sukaesih, N. A. Husin, A. Agustina, and Naghmeh Mahmoodian. "Sequential pattern mining on library transaction data." In Information Technology (ITSim), 2010 International Symposium in, vol. 1, pp. 1-4. IEEE, 2010.

[7] Srikant R. and Agrawal R. , "Mining sequential patterns: Generalization and performance improvements", Proceedings of the 5th International Conference Extending Database Technology, 1996, 1057, 3-17.

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

[9] S Vijayarani and S Deepa, "Sequential Pattern Mining A-Study". IJCA Proceedings on International Conference on Research Trends in Computer Technologies ICRTCT(1):14-18, February 2013.

[10] Chetna Chand, Amit Thakkar, and Amit Ganatra- "Sequential Pattern Mining: Survey and Current Research Challenges", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012.

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

[12] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of International Conference Very Large Data Bases (VLDB '94), pp. 487-499, Sept. 1994.

[13] Mallick Bhawna, Deepak Garg, and Preetam Singh Grover. "Constraint-Based Sequential Pattern Mining: A Pattern Growth Algorithm Incorporating Compactness, Length and Monetary." International Arab Journal of Information Technology, Vol. 11, No. 1 (2011).

[14] Chen, Yen-Liang, Mi-Hao Kuo, Shin-Yi Wu, and Kwei Tang. "Discovering recency, frequency, and monetary (RFM) sequential patterns from customers' purchasing data." Electronic Commerce Research and Applications 8, no. 5 (2009): 241-251.

[15] Boghey R. and Singh S., "Sequential Pattern Mining: A Survey on Approaches," IEEE International Conference on Communication Systems and Network Technologies (CSNT), vol., no., pp.670,674, Gwaliar,6-8 April 2013.

[16] Pei, Jian, Jiawei Han, and Wei Wang. "Constraint-based sequential pattern mining: the pattern-growth methods." Journal of Intelligent Information Systems 28, no. 2 (2007): 133-160.

[17] Morzy, Tadeusz, Marek Wojciechowski, and Maciej Zakrzewicz. "Efficient Constraint-Based Sequential Pattern Mining Using Dataset Filtering Techniques." In Databases and Information Systems II, pp. 297-309. Springer Netherlands, 2002.