Cent. Eur. J. Comp. Sci. • 2(4) • 2012 • 426-439
DOI: 10.2478/s13537-012-0030-8

VERSITA

Central European Journal of **Computer Science**

DRL-Prefixspan: A novel pattern growth algorithm for discovering downturn, revision and launch (DRL) sequential patterns

Research Article

Aloysius George*, D. Binu†

Department of Research & Development, Aloy labs, India

Received 26 March 2012; accepted 20 September 2012

Abstract: Discovering sequential patterns is a rather well-studied area in data mining and has been found many diverse applications, such as basket analysis, telecommunications, etc. In this article, we propose an efficient algorithm that incorporates constraints and promotion-based marketing scenarios for the mining of valuable sequential patterns. Incorporating specific constraints into the sequential mining process has enabled the discovery of more user-centered patterns. We move one step ahead and integrate three significant marketing scenarios for mining promotion-oriented sequential patterns. The promotion-based market scenarios considered in the proposed research are 1) product Downturn, 2) product Revision and 3) product Launch (DRL). Each of these scenarios is characterized by distinct item and adjacency constraints. We have developed a novel DRL-PrefixSpan algorithm (tailored form of the PrefixSpan) for mining all length DRL patterns. The proposed algorithm has been validated on synthetic sequential databases. The experimental results demonstrate the effectiveness of incorporating the promotion-based marketing scenarios in the sequential pattern mining process.

Keywords: sequential pattern mining • sequential database • PrefixSpan • projected database • constraint • adjacency • DRL-PrefixSpan • DRL-sequential pattern • product downturn • product revision • product launch • product life cycle

1. Introduction

Data mining has attracted a great deal of attention in the information industry in recent years due to the wide availability of huge amounts of data and imminent need for turning such data into useful information and knowledge [12]. Data mining is a part of the overall process of Knowledge Discovery in Databases (KDD). The accessibility and abundance of information makes data mining a matter of considerable importance and necessity [20]. Data mining has been defined as the non-trivial extraction of implicit, previously unknown and potentially useful information from data [6]. The development of data mining techniques has focused on efficiently discovering hidden information from large databases

* *E-mail: aloysiusgeorgephd@gmail.com*
† *E-mail: altimatebinu@gmail.com*

that is useful for corporate decision-makers [16, 17]. The data mining techniques include association rules mining, classification, clustering, mining time series, and sequential pattern mining, to name a few [7, 10, 13]. Among others, finding sequential patterns has attracted a significant amount of research attention [15].

Sequential pattern is a sequence of itemsets that frequently occur in a specific order, and all items in the same itemset are supposed to have the same transaction [33]. Each sequence corresponds to a temporally ordered list of events, where each event is a collection of items (itemset) occurring simultaneously. The temporal ordering among the events is induced by the absolute timestamps associated with the events [24]. Usually, all the transactions of a customer are together viewed as a sequence, called customer-sequence, where each transaction is represented as an itemset in that sequence and all the transactions are listed in a certain order with regard to the transaction-time [29]. The mining process of sequential pattern is very difficult and time-consuming due to several factors. First, the formation of a pattern is not limited to single items but itemsets. Second, neither the number of itemsets in a pattern nor the number of items in an itemset is known a priori. Third, patterns could be formed by any permutation, of any combination of possible items in the database [19].

Sequential pattern mining [22] is an important data-mining method for determining time-related behavior in sequence databases [8]. The sequential pattern mining was first introduced by Agrawal and Srikant in [1]: Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given a user-specified min_support threshold, sequential pattern mining is to find all frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is no less than min_support [25]. In their subsequent work, Agrawal et al. have discussed the introduction of constraints to the mined sequences, and have proposed an improved algorithm, GSP [32]. Following their work, many innovative and efficient algorithms have been presented for effective sequential pattern mining in the recent years. Sequential pattern mining algorithms, in general, can be categorized into three classes [27]:

- Apriori-based [1], horizontal formatting method, with GSP Srikant and Agrawal [32] as its representative;
- Apriori-based [1], vertical formatting method, such as SPADE [34]
- Projection-based pattern growth method, such as PSP [26] and SPAM [4].

In general, several sequential pattern mining algorithms have been designed for discovering sequential patterns. In recent years, researchers have recognized that frequency is not the best measure to use in determining the significance of a pattern in many applications. When using the single frequency constraint, the traditional mining method often generates a huge number of patterns and rules, but most of them are not useful. Due to its inefficiency and ineffectiveness, the importance of constraint-based pattern mining has been emphasized [8]. The inability to focus the discovery process on user expectations and background knowledge, has lead to a process that is, in many cases, prohibitively expensive and very difficult to deal. The treatment of sequential data (the analysis of frequent behaviors, for example) is a particular case of pattern mining, usually known as sequential pattern mining, and suffers from the same drawbacks. In order to minimize this problem, sequential pattern mining algorithms use constraints to restrict the number and scope of discovered patterns [3].

In constraint-based sequential pattern mining, the following classes of constraints are identified: database constraints, pattern constraints, and time constraints. Database constraints are used to specify the source dataset. Pattern constraints specify which patterns are interesting and should be returned by the query. Finally, time constraints influence the process of checking whether a given data-sequence contains a given pattern [23]. Constraints make it possible to focus the mining process into areas or sub-spaces where useful information is likely to be gained [3, 5]. It is obvious that additional constraints can be verified in a post-processing step, after all patterns exceeding a given minimum support threshold have been discovered. Nevertheless, such a solution cannot be considered satisfactory since users providing advanced pattern selection criteria may expect that the data mining system will exploit them in the mining process to improve performance. It has been shown that pushing constraints deep into the mining process can reduce processing time by more than an order of magnitude [11, 21, 23].

In this paper, we incorporate two constraints, namely item and adjacency in addition to frequency, for discovering interesting and valuable sequential patterns from sequential databases. Incorporating the constraints and the marketing scenarios into the original PrefixSpan algorithm [25], we develop a novel DRL-PrefixSpan algorithm for generating all DRL sequential patterns from the sequential database. The DRL-PrefixSpan algorithm discovers the complete set of patterns employing a divide-and-conquer strategy. The novel algorithm first mines the 1-DRL patterns by considering the

DRL scenarios and then builds upon the projected database corresponding to the mined 1-DRL patterns. Subsequently, 2-DRL patterns are mined from the projected database and the process is applied recursively until all length DRL patterns are mined. The discovered DRL sequential patterns signify valuable information on customer purchasing behavior and would be a good indicator for promotion-based managerial decision-making.
The main contribution of the paper is given as,

- Develop a novel DRL-PrefixSpan algorithm to generate DRL sequential patterns from the sequential database.
- Consider three promotion-based conditional scenarios presented in a dynamic marketing environment namely, product Downturn, product Revision and product Launch (DRL) in the sequential pattern mining process.
- Analyze the novel algorithm with the synthetic data to prove the performance.

The rest of the paper is organized as follows: Related research is discussed in Section 2. The problem statement and abstract algorithm are given in Section 3. The proposed constraint-based sequential pattern mining algorithm for handling marketing uncertainties is presented in Section 4. The experimental result of the proposed algorithm is given in Section 5. The conclusion is made in Section 6.

2. Review of related papers

A significant number of techniques for mining sequential patterns from a database are available in the literature. Here, candidate sequence and projection-based methods are widely used for mining sequential patterns. But, Ming-Yen Lin and Suh-Yin Lee [18] have proposed DELISP (delimited sequential pattern) approach in reducing the size of projected databases by bounded and windowed projection techniques. Bounded projection keeps only time-gap valid subsequences and windowed projection saves non-redundant subsequences by satisfying the sliding time-window constraint. Furthermore, the delimited growth technique directly generated constraint-satisfactory patterns and sped up the pattern growing process. The comprehensive experiments conducted showed that DELISP has good scalability and outperformed the well-known GSP algorithm in the discovery of sequential patterns with time constraints.
Jian Pei et al. [27] have developed a framework for constraint-based sequential pattern mining. Their study showed that constraints can be effectively and efficiently pushed deep into the sequential pattern mining under the framework. Moreover, the framework can be extended to constraint-based structured pattern mining as well. After that, time constrains are effectively incorporated into the sequential pattern mining algorithm. Accordingly, Ya-Han Hu et al. [14] have developed two efficient algorithms, called the MI-Apriori and MI-PrefixSpan algorithms to mine a variant of time-interval sequential patterns, called multi-time-interval sequential patterns, which revealed the time-intervals between all pairs of items in a pattern. The experimental results showed that the MI-PrefixSpan algorithm was faster than the MI-Apriori algorithm, but the MI-Apriori algorithm has better scalability in long sequence data.
Again, the problem of discovering sequential patterns by handling time constraints is done by F. Masseglia et al. [21] who, proposed an efficient algorithm, called GTC (Graph for Time Constraints) for mining such patterns in very large databases. One of the most significant features of the approach was that handling of time constraints can be easily taken into account in traditional level-wise approaches since it was carried out prior to and separately from the counting step of a data sequence (frequency computation). Their test showed that the algorithm performed significantly faster than a state-of-the-art sequence mining algorithm.
Instead of time constraints, some of the authors utilized the multiple constraints into the sequential pattern mining framework. Similar type of research works have been discussed in [8, 9, 30]. Jiadong Ren et al. [30] have incorporated recency and compactness constraints for frequent patterns mining. To mine more efficiently in the incremental database, two concepts of recency and compactness were introduced into sliding-window filtering (denoted as SWF). By employing SWF with constraints of compactness and recency, user satisfactory CFR-patterns (compactness, frequency and recency) were discovered. Yen-Liang Chen and Ya-Han Hu [8] have developed a CFR-PostfixSpan algorithm. They incorporated two concepts namely, recency and compactness, into pattern growth methodology.
Yen-Liang Chen et al. [9] have incorporated the recency, frequency, and monetary (RFM) concept presented in the marketing literature to define the RFM sequential pattern and developed an algorithm for generating all RFM sequential patterns from customers' purchasing data. Using the algorithm, they proposed a pattern segmentation framework to generate valuable information on customer purchasing behavior for managerial decision-making. The major difference

between the proposed work with the works available in [8, 9, 30] is that the proposed work utilized the "product life cycle"-based conditional scenarios as three constraints in mining sequential behavior using pattern growth methodology. In the existing works [8, 9, 30], they have used the compactness, monetary as well as recency constraints in mining sequential patterns using pattern growth methodology.

3. Problem statement and abstract algorithm

3.1. Sequential pattern

The sequential pattern mining problem is to find the complete set of sequential patterns with respect to a given sequence database S and a support thresholdmin _ sup.
Let $I = \{z_1, \dots, z_n\}$ 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 $\alpha = \langle Z_1 \dots Z_l \rangle$ is an ordered list of item-sets. An itemset Z_i $(1 \le i \le l)$ in a sequence is called a transaction, a term originated from analyzing customers' shopping sequences in a transaction database, such as in [31]. A transaction Z_i may have a special attribute, time-stamp, denoted as $Z_i.\ t$, which registers the time when the transaction was executed. As a notational convention, for a sequence $\alpha = \langle Z_1 \cdots Z_l \rangle, Z_i.\ t < Z_j.\ t$ for $1 \le \mathrm{i} < \mathrm{j} \le \mathrm{l}$., the number of transactions in a sequence is called the length of the sequence. A sequence α with length l is called an l-sequence, denoted as $len(\alpha) = l$. A sequence $\alpha = \langle Z_1 \dots Z_n \rangle$ is called a subsequence of another sequence $\beta = \langle Y_1 \cdots Y_m \rangle$, $(n \le m)$ and β a super-sequence of α, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \le i_1 < \cdots < i_n \le m$ such that $Z_1 \subseteq Y_{i_1}, \cdots, Z_n \subseteq Y_{i_n}$. A sequence database S is a set of 2-tuples (sid, α), where sid is a sequence-id and α a sequence. A tuple (sid, α) in a sequence database S is said to contain a sequence γ if γ is a subsequence of α. The number of tuples in a sequence database S containing sequence γ is called the support of γ, denoted as $\sup(\gamma)$. Given a positive integer min _ sup as the support threshold, a sequence γ is a sequential pattern in sequence database S if $\sup(\gamma) \ge$ min _ sup [28].

3.2. Constrained sequential pattern

Despite the conceptual simplicity of sequential pattern mining, the existing approaches suffer from two major drawbacks.
Disproportionate computational cost for selective users: Given a database of sequences and a fixed value for the minimum support threshold, the computational cost of the pattern mining process is fixed for any potential user. Ignoring user focus can be extremely unfair to a highly selective user that is only interested in patterns of a very specific form.
Overwhelming volume of potentially useless results: The lack of tools to express user focus during the patter mining process means that selective users will typically be swamped with a huge number of frequent patterns, most of which are useless for their purposes.
The above discussion clearly demonstrates the need for novel pattern mining solutions that enable the incorporation of user-controlled focus in the mining process. The two main constraints that signify the user need are,
Constraint 1 (Item constraint): An item constraint specifies subset of items that should or should not be present in the patterns. It is in the form of

cu, or

$$C_{item}\ (\alpha) \equiv (\Psi\ \mathrm{i} : 1 \le \mathrm{i} \le len\,(\alpha),\ \alpha[i] \cap \mathrm{V} \neq\ \phi),$$

where V is a subset of items, $\Psi\ \in \{\forall, \exists\}$ and $\theta\ \in\ \{\subseteq, \not\subseteq, \supseteq, \not\supseteq, \in, \notin\}$. For the sake of brevity, we omit the strict operators (e.g., $\subset$, $\supset$) in our discussion here. However, the same principles can be applied to them.
Constraint 2 (Adjacency constraint): An adjacency constraint is defined only in a sequence database where each transaction in every sequence has a time-stamp. It requires that the sequential patterns in the sequence database must have the property such that the time-stamp of item in the sequential pattern must not be greater or lesser than a predefined threshold.
Formally, an adjacency constraint is in the form of,

$$C_{dur} \equiv Adj\,(\alpha)\ \theta\ t,$$

where $\theta\ \in \{\le, \ge\}$ and t is a given integer.

3.3. Promotion-based conditional scenarios

Businesses should manage their products carefully over time to ensure that they deliver products that continue to meet customer needs. The process of managing groups of brands and product lines is called portfolio planning.
The stages through which individual products develop over time are commonly called the "Product Life Cycle". The classic product life cycle has four stages (illustrated in the Figure 1: introduction; growth; maturity and decline.

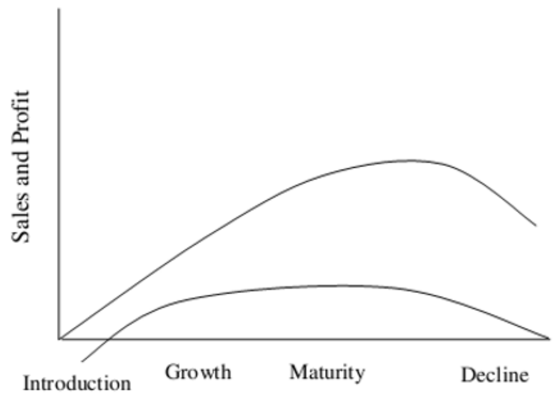


Figure 1. Product Life Cycle.

Introduction Stage: At the Introduction (or development) Stage market size and growth is slight. Products at this stage have to be carefully monitored to ensure that they start to grow.
Growth Stage: The Growth Stage is characterized by rapid growth in sales and profits.
Maturity Stage: The Maturity Stage is, perhaps, the most common stage for all markets. It is aimed towards product modification and improvement so as to improve production efficiency and quality.
Decline Stage: In the Decline Stage, the market is shrinking, reducing the overall amount of profit that can be shared amongst the remaining competitors.

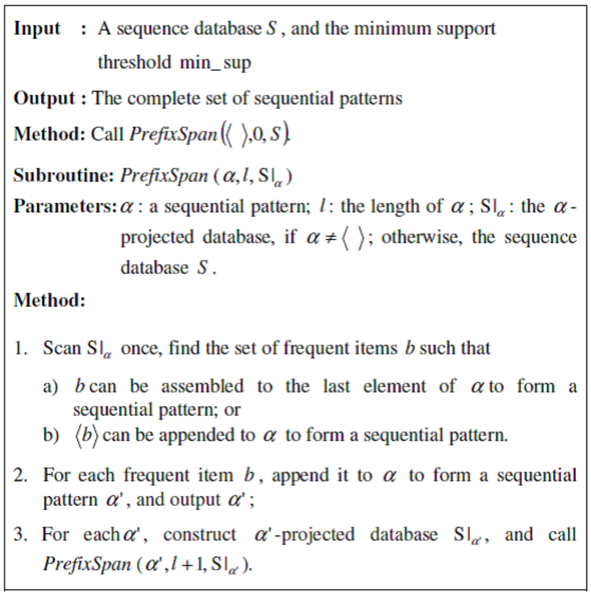
Introduction	Growth	Maturity	Decline
Third generation mobile phones	Portable DVD Players	Personal Computers	Typewriters
Iris-based personal identity cards	Smart cards	Credit cards	Cheque books

The stages in the product life cycle are generally initiated by three possible cases, downturn of products in the market, release of incremental products and launch of new or breakthrough products.
Outdated products are archaic products that are of least value to the business and not anymore the taste of the clients. Mining sequences containing archaic items is not going to improve the productivity of the business.
Incremental products are generally considered to be cost reductions, improvements to existing product lines, additions to existing platforms and repositioning of existing products introduced in markets well known to the company, with well identified customer needs using technology in which the company already has expertise. Incremental products, if successful, gradually decline the existing product lines, and if not, they gradually end up in the market being unable to catch-up with the existing product lines. So, hereby it is more intricate to emphasize either on the incremental product or the existing product lines.
Breakthrough products (i.e. new to the company or new the world) typically begin either with a strategic vision or are identified and persevered by an individual/product champion. The breakthrough products have to be carefully monitored and nurtured to ensure that they start to grow.

3.4. Sequence mining algorithm

PrefixSpan is the most promising of the pattern-growth methods and is based on recursively constructing the patterns. In general, several algorithms have been developed based on Apriori (E.g. GSP algorithm) and pattern growth (E.g.

PrefixSpan algorithm) method for effective sequential pattern mining. Apriori-based algorithms encounters some problems such as, (1) Potentially huge set of candidate sequences, (2) Multiple scans of database, (3) Difficulties at mining long sequential patterns. PrefixSpan algorithm was developed to solve the problems encountered by the Apriori-based methods. The idea behind the Pattern-growth is to avoid the candidate generation step altogether, and to focus the search on a restricted portion of the initial database [2]. The PrefixSpan algorithm is shown in Figure 2.



```
Input   : A sequence database $S$, and the minimum support
          threshold min_sup
Output : The complete set of sequential patterns
Method: Call $PrefixSpan(\langle\ \rangle, 0, S)$

Subroutine: $PrefixSpan\ (\alpha, l, S|_{\alpha})$
Parameters: $\alpha$ : a sequential pattern; $l$ : the length of $\alpha$ ; $S|_{\alpha}$ : the $\alpha$-
            projected database, if $\alpha \neq \langle\ \rangle$; otherwise, the sequence
            database $S$.
Method:

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

Figure 2. PrefixSpan algorithm.

4. Proposed constraint-based sequential pattern mining algorithm for handling marketing uncertainties

Constraints, which represent user's interest and focus, are useful for discovering the interesting and useful patterns that provide valuable information for improving the business. In analyzing business promotion, it is necessary to understand the fact that current purchase behavior of customers is a better indicator for effectual discovery of sequential patterns rather than the entire set of customer buying patterns. Constraints facilitate the aforesaid scenario by prioritizing those high impact patterns essential for business promotion. In the proposed algorithm, we make use of the item and timing constraints to provide a greater emphasis on the recent purchasing trend of the customers. The constraints are named as follows:

- Item
- Adjacency

The constraints such as item and adjacency are closely related to each other and also interdependent. The itemset to be discarded or included into the sequence is decided by the timestamp associated with it (adjacency). There have been papers in the literature that perform sequential pattern mining based on item and timing constraints. But, there has not been many works that perform sequential pattern mining by taking into account the various marketing uncertainties and scenarios. This is chiefly because of the dynamicity associated with the marketing environment. Here, we consider the marketing scenarios caused by product downturn, product revision and product launch. The products corresponding to each of the scenarios are termed as outdated products, incremental products and breakthrough products. The following set of definitions will provide good insight into the proposed approach for sequential pattern mining,
Definition 1 (Product downturn): It refers to the set of outdated products that are of least significance to the business. The product should be eliminated when it no longer provides a strategic, economic, or competitive advantage for the company.

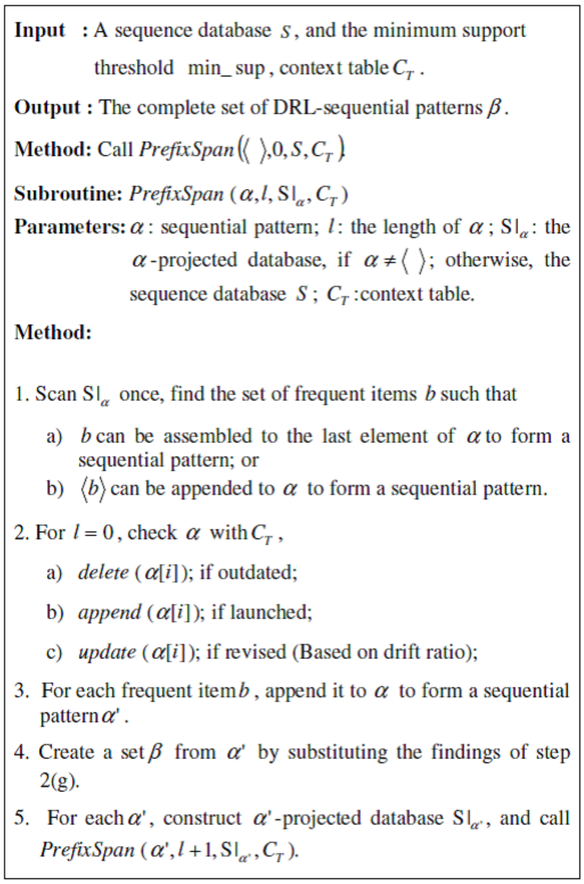
Definition 2 (Product revision): It refers to the set of incremental products that amend the existing product lines. The incremental product or the existing product line should be eliminated by determining a drift ratio that measures the significance of the product in the market.
Definition 3 (Product Launch): It refers to the set of breakthrough products that are of future significance to the business. The product should be included irrespective of its strategic, economic, or competitive advantage for the company.
Definition 4 (Drift ratio): Drift ratio is the measure of the relative significance of the incremental item to the existing product line. It can be defined as

$$D_g = \frac{C_2 - C_1'}{C_1}$$

where, $C_2 \rightarrow$ Frequency of the incremental product in D.
$C_1' \rightarrow$ Frequency of the existing product line after product revision.
$C_1 \rightarrow$ Frequency of the existing product line before product revision.
$D_g \rightarrow$ Drift ratio.

4.1. DRL-PrefixSpan algorithm

We incorporate the item and the adjacency constraints into the PrefixSpan algorithm and the promotion-based marketing scenarios namely product Downturn, product Revision and product Launch, to arrive at significant DRL-sequential patterns. The DRL-PrefixSpan algorithm (shown in Figure 3) is an extension of the well-known PrefixSpan algorithm.



```
Input  : A sequence database $S$, and the minimum support
         threshold $min\_sup$, context table $C_T$.
Output : The complete set of DRL-sequential patterns $\beta$.
Method: Call $PrefixSpan(\langle\ \rangle, 0, S, C_T)$

Subroutine: $PrefixSpan\ (\alpha, l, S|_\alpha, C_T)$
Parameters: $\alpha$: sequential pattern; $l$: the length of $\alpha$; $S|_\alpha$: the
            $\alpha$-projected database, if $\alpha \neq \langle\ \rangle$; otherwise, the
            sequence database $S$; $C_T$ :context table.
Method:

1. Scan $S|_\alpha$ once, find the set of frequent items $b$ such that
    a) $b$ can be assembled to the last element of $\alpha$ to form a
       sequential pattern; or
    b) $\langle b \rangle$ can be appended to $\alpha$ to form a sequential pattern.
2. For $l = 0$, check $\alpha$ with $C_T$,
    a) delete ($\alpha[i]$); if outdated;
    b) append ($\alpha[i]$); if launched;
    c) update ($\alpha[i]$); if revised (Based on drift ratio);
3. For each frequent item $b$, append it to $\alpha$ to form a sequential
   pattern $\alpha'$.
4. Create a set $\beta$ from $\alpha'$ by substituting the findings of step
   2(g).
5. For each $\alpha'$, construct $\alpha'$-projected database $S|_{\alpha'}$, and call
   $PrefixSpan\ (\alpha', l+1, S|_{\alpha'}, C_T)$.
```

Figure 3. DRL-PrefixSpan algorithm.

The major steps of the DRL-PrefixSpan algorithm are given as follows:(1) Find 1-DRL patterns, (2) Divide search space, and (3) Find subsets of sequential patterns. Each step is explained in further detail below.

Step 1: Find 1-DRL patterns

We mine 1-DRL pattern (1-length DRL patterns) from the sequential database by scanning the database once. The patterns mined are tailored with respect to the three different marketing scenarios (Product downturn, Product revision and Product launch). Each of these scenarios correspond to different timing constraints and item constraints contained in the context table C_T. The context table $C_T$contains the adjacency and the entity information associated with the individual items in the sequential database D. For determining 1-DRL patterns, the algorithm checks with C_T and,

- If the item is outdated *(Def. 1)*, exclude the item from the 1-DRL patterns generated.
- For a breakthrough item *(Def. 3)*, include the item into 1-DRL patterns generated.
- For incremental items *(Def. 2 & 4)*, the incremental or the existing product lines are included or excluded based on the drift equation.

$$D_g = \frac{C_2 - C_1'}{C_1}$$

Example 1: Let D be a sequence database (Table 1) and C_T be the context table (Table 2). We scan the sequence database D once to count the support of individual items present in it [(a→5), (b→5), (c→4), (d→5), (e→5), (f→4), (g→3)]. The set of 1-length frequent patterns generated with $\min_sup = 3$is {a, b, c, d, e, f, g}. FromC_T, we determine that, ‘g’ is an outdated product, ‘d’ is a breakthrough product, ‘e’ and ‘f ’ are the incremental products of existing product lines ‘a’ and ‘b’ respectively. So, the outdated product ‘g’ is excluded from the generated 1-length frequent patterns and breakthrough product‘d’ must be included into the 1-length frequent patterns irrespective of its frequency. Regarding incremental products, the drift ratio D_g of ‘e’ w.r.t ‘a’ and ‘f’ w.r.t ‘b’ are computed. The values obtained are 1 and -1.5 respectively. From the values of D_g, the incremental product ‘e’ is included into the 1-length frequent patterns instead of ‘a’ and ‘f’ is not included for ‘b’. The pattern set to be utilized for projection i.e. projection set, is {a, b, c, d, e}. The actual set of 1-DRL patterns obtained is {b, c, d, e}.

Table 1. Sequential database.

SID	Sequence
10	<(b,1), (b,4), (a,4), (e,5), (b,5), (e,6), (a,6), (d,7), (e,7), (f,7), (e,8), (b,8)>
20	<(g,1), (a,4), (b,5), (c,5), (g,5), (d,6), (e,6), (f,8), (b,8)>
30	<(c,1), (a,5), (b,5), (b,6), (e,7), (b,7), (d,8), (e,8)>
40	<(a,1), (b,1), (g,3), (b,3), (a,4), (b,4), (e,5), (c,5), (a,7), (d,8), (f,8)>
50	<(a,1), (c,1), (b,2), (c,2), (g,3), (a,4), (b,5), (e,6), (d,8), (f,8)>

Table 2. Context table.

Product Type	Product (timestamp)
Outdated Product	$\langle g\rangle$ (t_1- t_5)
Existing product → Incremental Product	$\langle a\rangle$(t_1- t_8) $\langle e\rangle$(t_5- t_8), $\langle b\rangle$(t_1- t_8) $\langle f\rangle$(t_7- t_8)
Breakthrough Product	$\langle d\rangle$(t_6- t_8)

Step 2: Divide search space

The projection set is then used to construct the projected database, which consists of non-empty (postfix) subsequences having patterns in the projection set as their prefix. Let $a_1, a_2, \ldots, a_n$ be the complete set of one length patterns in the projection set. We can obtain n disjoint subsets from the complete set of 1-length patterns in the projection set. The ith subset $(1 \leq i \leq n)$is the set of sequential patterns with prefix a_i.

Example 2: We partition the database into five subsets: (1) with prefix ‘a’ (a-projected database), (2) with prefix ‘b’ (b-projected database), (3) with prefix ‘c’ (c-projected database), (4) with prefix ‘d’ (d-projected database), and (5) with prefix ‘e’ (c-projected database). The steps involved in building a a-projected database are, for the first data sequence

⟨(b,1), (b,4), (a,4), (e,5), (b,5), (e,6), (a,6), (d,7), (e,7), (f,7), (e,8), (b,8)⟩, the item 'a' has the timestamp 4, and then, the projection w.r.t. item 'a' is ⟨(e,5), (b,5), (e,6), (a,6), (d,7), (e,7), (f,7), (e,8), (b,8)⟩. In the second data sequence ⟨(g,1), (a,4), (b,5), (c,5), (g,5), (d,6), (e,6), (f,8), (b,8)⟩, item 'a' has the timestamp 4 and the projection w.r.t. item a is ⟨(b,5), (c,5), (g,5), (d,6), (e,6), (f,8), (b,8)⟩. Similarly, we project for the entire sequence and finally yielding a's projected databases. The aforesaid procedure is repeated to build the projected database for all other patterns in the projection set. Table 3 shows the projected database of all one length patterns in the projection set.

Table 3. Projected database for 1-DRL pattern.

<a>	<(e,5), (b,5), (e,6), (a,6), (d,7), (e,7), (f,7), (e,8), (b,8)> <(b,5), (c,5), (g,5), (d,6), (e,6), (f,8), (b,8)> <(b,5), (b,6), (e,7), (b,7), (d,8), (e,8)> <(b,1), (g,3), (b,3), (a,4), (b,4), (e,5), (c,5), (a,7), (d,8), (f,8)> <(c,1), (b,2), (c,2), (g,3), (a,4), (b,5), (e,6), (d,8), (f,8)>
<b>	<(b,4), (a,4), (e,5), (b,5), (e,6), (a,6), (d,7), (e,7), (f,7), (e,8), (b,8)> <(c,5), (g,5), (d,6), (e,6), (f,8), (b,8)> <(b,6), (e,7), (b,7), (d,8), (e,8)> <(g,3), (b,3), (a,4), (b,4), (e,5), (c,5), (a,7), (d,8), (f,8)> <(c,2), (g,3), (a,4), (b,5), (e,6), (d,8), (f,8)>
<c>	<(g,5), (d,6), (e,6), (f,8), (b,8)> <(a,5), (b,5), (b,6), (e,7), (b,7), (d,8), (e,8)> <(a,7), (d,8), (f,8)> <(b,2), (c,2), (g,3), (a,4), (b,5), (e,6), (d,8), (f,8)>
<d>	<(e,7), (f,7), (e,8), (b,8)> <(e,6), (f,8), (b,8)> <(f,8)> <(f,8)>
<e>	<(b,5), (e,6), (a,6), (d,7), (e,7), (f,7), (e,8), (b,8)> <(f,8), (b,8)> <(b,7), <d,8>, (e,8)> <(c,5), (a,7), (d,8), (f,8)> <(d,8), (f,8)>

Step 3: Find subsets of sequential patterns

The subsets of 1-DRL patterns can be mined from the corresponding set of projected databases and the process is done recursively. The mined DRL patterns are shown in Table 4.

Table 4. DRL sequential patterns.

<a>	<e>, <eb>, <ebb>, <ebbe>, <ebbd>, <ebe>, <ebee>, <ebed>, <ebef>, <ebe(df)>, <ebd>, <ebf>, <eb(de)>, <eb(df)>, <ee>, <eee>, <eed>, <eef>, <ee(df)>, <ed>, <ef>, <e(de)>, <e(df)>, <ec>, <ecd>, <ecf>
<b>	<b>, <bb>, <bbb>, <bbbe>, <bbbd>, <bbe>, <bbed>, <bbef>, <bbe(df)>, <bbd>, <bbf>, <bb(df)>, <ba>, <bae>, <baed>, <baef>, <bae(df)>, <bad>, <baf>, <ba(df)>, <be>, <bed>, <bef>, <be(df)>, <bd>, <bf>, <b(de)>, <b(df)>
<c>	<c>, <cb>, <ca>, <cad>, <ce>, <cd>, <cf>
<d>	<d>, <(de)>, <(df)>
<e>	<e>, <ed>, <ef>, <e(df)>

Example 3: The projected databases for all qualified 1-length patterns are listed in Table 3 and the mining process is discussed as follows. First, we discover the DRL patterns having prefix $\langle a \rangle$. The sequences containing prefix $\langle a \rangle$should be identified from the $\langle a \rangle$projected database. By scanning the $\langle a \rangle$-projected database once, we determine the locally frequent items namely, a →3, b → 5, c → 3, d → 4 and e → 5. Thus, all the 2-length sequential patterns prefixed with

$\langle a \rangle$ are discovered from the $\langle a \rangle$-projected database. They are: {aa, ab, ac, ad, ae}. Since, 'a' is the existing product line of the incremental product 'e', by prefix drift property, we replace 'a' with 'e'. The 2-DRL patterns thus obtained are {ee, eb, ec, ed, ee}. Again, $\langle a \rangle$-projected database can be divided into 5 subsets: (1) those with prefix $\langle aa \rangle$, (2) those with prefix $\langle ab \rangle$, (3) those with prefix $\langle ac \rangle$, (4) those with prefix $\langle ad \rangle$, and (5) those with prefix $\langle ae \rangle$. The 3-DRL patterns with prefix$\langle aa \rangle$, $\langle ab \rangle$, $\langle ac \rangle$, $\langle ad \rangle$, $\langle ae \rangle$ are mined from the respective projected databases by scanning the database once and the prefix drift property is applied to each of the patterns. Recursively, we do this process to discover all length DRL patterns with prefix$\langle a \rangle$. The above procedure is repeated for other 1-DRL patterns$\langle b \rangle$, $\langle c \rangle$, $\langle d \rangle$ and $\langle e \rangle$. The mined DRL-patterns are shown in Table 4.

5. Results and discussion

The results and the discussion of the proposed DRL-PrefixSpan algorithm for effectual sequential pattern mining are presented in this section. The DRL-PrefixSpan algorithm has been implemented in Java (jdk 1.6).

5.1. Experimental results

The sample sequential database taken for experimentation is given in Table 1. The context table containing entity and adjacency information presented in Table 2. The projected database for the 1-DRL patterns is given in Table 3. The discovered DRL patterns are given in Table 4. The sequential patterns mined by the actual PrefixSpan algorithm are given in Table 5. The number of sequential patterns generated for different pattern lengths using the PrefixSpan algorithm and DRL- PrefixSpan algorithm is presented in Table 6.

Table 5. Sequential pattern discovered by PrefixSpan algorithm.

<a>	<a>, <ab>, <abb>, <abbe>, <abbd>, <aba>, <abae>, <abad>, <abaf>, <aba(df)>, <abe>, <abed>, <abef>, <abe(df)>, <abd>, <abf>, <ab(de)>, <ab(df)>, <aa>, <aae>, <aad>, <aaf>, <aa(df)>, <ae>, <aed>, <aef>, <ae(df)>, <ad>, <af>, <a(de)>, <a(df)>, <ac>, <acd>, <acf>, <ag>, <agb>, <age>, <agef>, <agd>, <agf>
<b>	<b>, <bb>, <bbb>, <bbbe>, <bbbd>, <bbe>, <bbed>, <bbef>, <bbe(df)>,<bbd>, <bbf>, <bb(df)>, <ba>, <bae>, <baed>, <baef>, <bae(df)>, <bad>, <baf>, <ba(df)>, <be>, <bed>, <bef>, <be(df)>, <bd>, <bf>, <b(de)>, <b(df)>
<c>	<c>, <cb>, <ca>, <cad>, <ce>, <cd>, <cf>
<d>	<d>, <(de)>, <(df)>
<e>	<e>, <ed>, <ef>, <e(df)>
<f>	<f>
<g>	<g>, <gb>, <ge>, <gef>, <gd>, <gf>

Table 6. Number of sequential patterns generated by PrefixSpan and DRL- PrefixSpan algorithm.

Length of sequential pattern	Number of sequential patterns mined	
	PrefixSpan Algorithm	DRL-PrefixSpan Algorithm
1	7	5
2	25	19
3	31	24
4	21	17
5	4	3

5.2. Performance analysis

The performance of the novel algorithm of mining sequential pattern from the database is analyzed using the synthetic data. The synthetic data used in the performance analysis contains 25,000 sequences of records with 16 items. We use two measures for performance evaluation: (1) number of sequential patterns mined and (2) computation time.

Number of sequential patterns mined: Initially, input data comprising 25,000 sequences are given to the DRL-prefixspan algorithm which produces a number of sequential patterns. The number of sequential patterns generated from Prefixspan and DRL-prefixspan algorithm is computed by inputting the various support values as a percentage and the graph is plotted in Figure 4. From Figure 4, we can analyze that the number for patterns mined for proposed algorithm is less than with the traditional algorithm. Also, the number of patterns mined from the database is reduced whenever the support value is increased. For the support percentage of 50, the novel algorithm produced only 9100 patterns which is less than with the patterns produced by the PrefixSpan algorithm (16,200). This signifies that the constraint patterns are mined from the sequential database using the DRL-PrefixSpan algorithm.

The scalability issue in terms of number patterns mined is analyzed in Figure 5. Here, the graph is plotted for the various number of sequence records in the database. For the various numbers of records, the sequential patterns are mined for the support of 50% and the corresponding changes in the results are analyzed. Here, whenever the database size is increased, the number of patterns mined from the database is also increased slightly. In prefixspan, 2000 patterns are additionally obtained whenever the database size is increased from 10,000 to 25, 000. In addition, the proposed algorithm mined the same number of patterns for the different database sizes.

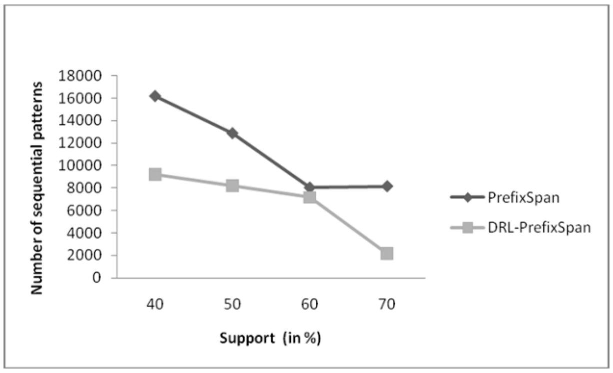


Figure 4. Comparison graph for various support values.

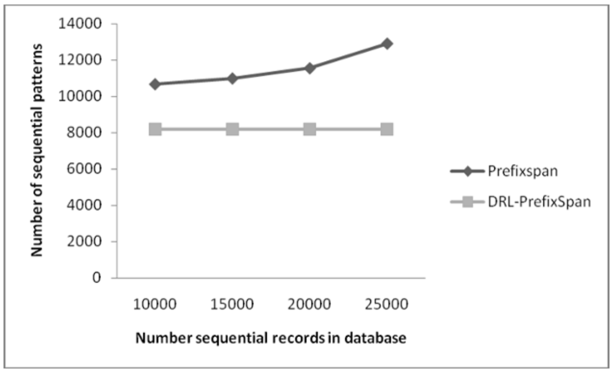


Figure 5. Comparison graph for various number records in database.

Computation time: The computation time of both algorithm is analyzed for the input data of having 25,000 sequences. For various support values, the computation time needed to mine the sequential patterns using PrefixSpan and DRL-

PrefixSpan algorithm is computed and the graph is plotted in Figure 6. From Figure 6, we can analyze that the computation time of proposed algorithm is less compared with the traditional algorithm. For the support percentage of 40, the novel algorithm takes 2700 seconds which is less than with the PrefixSpan algorithm (8,200 seconds).
The scalability issue for the various database sizes is evaluated in Figure 7. Here, for the different database sizes, the computation time is taken for the support of 50% and the performance in the results is analyzed. The computation time needed to mine patterns from the database is increased whenever the database size is also increased. The changes of proposed algorithm from 10,000 to 25, 000 is only 1800 seconds but, the traditional algorithm takes more time of 3000 seconds.

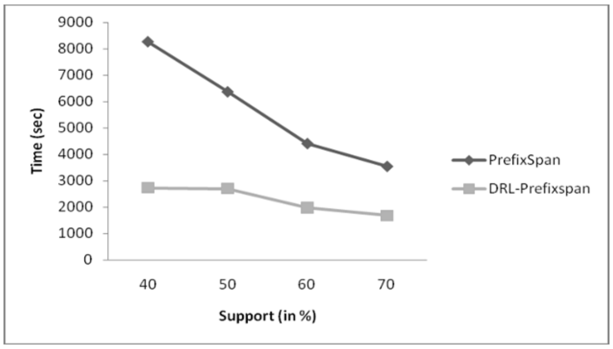


Figure 6. Comparison graph of computation time for various support values.

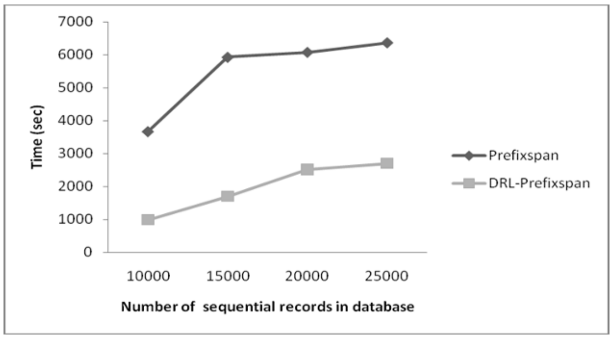


Figure 7. Comparison graph of computation for various number records in database.

6. Conclusion

We have developed a novel algorithm, DRL-PrefixSpan algorithm, for generating all DRL sequential patterns from the sequential database. The DRL-PrefixSpan algorithm is a pattern-growth methodology that discovers patterns by employing a divide-and-conquer strategy. We have used two constraints: namely, item and adjacency, in addition to frequency for discovering interesting and valuable sequential patterns from the sequential database. We have also considered the promotion-based marketing scenarios: product Downturn, product Revision and product Launch (DRL) to derive the sequential patterns. In the novel algorithm, the sequence database has been recursively projected into a set of smaller projected databases, and DRL patterns have been discovered in each projected database by exploring only locally frequent fragments. The experimental results have demonstrated the effectiveness of incorporating the promotion-based marketing scenarios in the sequential pattern mining process.

References

[1] Agrawal R., Srikant R., Mining Sequential Patterns, In: Proceedings of the International Conference on Data Engineering. (ICDE '95), 3-14, 1995

[2] Antunes C., Oliveira A.L., Generalization of Pattern-growth Methods for Sequential Pattern Mining with Gap Constraints, Lect. Notes Comput. Sci., 2734, 239-251, 2003

[3] Antunes C., Oliveira A.L., Sequential Pattern Mining With Approximated Constraints, In: Proceedings of the International Conference on Applied Computing, 131-138, 2004

[4] Ayres J., Flannick J., Gehrke J., T Y., Sequential pattern mining using a bitmap representation, In: Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD'02), 429-435, 2002

[5] Bayardo R.J., The Many Roles of Constraints in Data Mining, SIGKDD Explorations, 4, i-ii, 2002

[6] Bollmann-Sdorra P., Hafez A.M., Raghavan V.V., A Theoretical Framework for Association Mining based on the Boolean Retrieval Model, In: Proceedings of the International conference on Data warehousing and knowledge discovery, 2114, 21-30, 2001

[7] Chen M.-S., Han J., Yu P.S., Data Mining: An Overview from Database Perspective, IEEE Trans. Knowl. Data Eng., 8, 866-883, 1996

[8] Chen Y.-L., Hu Y.-H., Constraint-based sequential pattern mining: The consideration of recency and compactness, Decis. Support Systems, 42, 1203-1215, 2006

[9] Chen Y.-L., Kuo M.-H., Wu S.-Y., Tangc K., Discovering recency, frequency, and monetary (RFM) sequential patterns from customers' purchasing data, Electron. Commerce Res. Appl., 8, 241-251, 2009

[10] Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurasamy R., Advances in Knowledge Discovery and Data Mining (MIT Press, 1996)

[11] Garofalakis M., Rastogi R., Shim K., Mining Sequential Patterns with Regular Expression Constraints, IEEE Trans. Knowl. Data Eng., 14, 530-552, 2002

[12] Ghatari A.R., Mohamadi N., Honarmand A., Ahmadi P., Mohamadi N., Recognizing & Prioritizing Of Critical Success Factors (CSFs) On Data Mining Algorithm's Implementation In Banking Industry: Evidence From Banking Business System", In: Proceedings of EABR & TLC Conference, 2009

[13] Han J., Kamber M., Data Mining: Concepts and Techniques (Morgan Kaufmann, 2000)

[14] Hu Y.H., Huang T.Ch-K., Yang H.-R., Chen Y.-L., On mining multi-time-interval sequential patterns, Data & Knowledge Engineer., 68, 1112-1127, 2009

[15] Huang J.-W., Tseng Ch.-Y., Ou J.-Ch., Chen M.-S., A General Model for Sequential Pattern Mining with a Progressive Database, IEEE Trans. Knowl. Data Eng., 20, 2008

[16] Kantardzic M., Data Mining: Concepts, Models, Methods, and Algorithms (John Wiley & Sons Inc., 2002)

[17] Li Y.-Ch., Yeh J.-S., Chang Ch.-Ch., Isolated items discarding strategy for discovering high utility itemsets, Data & Knowledge Eng., 64, 198-217, January 2008

[18] Lin M.-Y., Lee S.-Y., Efficient mining of sequential patterns with time constraints by delimited pattern growth, Knowl. Inf. Syst., 7, 499-514, 2005

[19] Lin M.-Y., Lee S.-Y., Interactive Sequence Discovery by Incremental Mining, Int. J. Inf. Sci., 165, 187-205, 2004

[20] Maimon O.Z., Rokach L., Decomposition Methodology for Knowledge Discovery and Data Mining: Theory and Applications (World Scientific Publishing Company, 2005)

[21] Masseglia F., Poncelet P., Teisseire M., Efficient Mining Sequential Patterns with Time Constraints: Reducing the Combinations, Expert Syst. Appl., 36, 2677-2690, 2009

[22] Masseglia F., Poncelet P., Teisseire M., Incremental mining of sequential patterns in large databases, Data & Knowledge Engineer., 46, 97-121, 2003

[23] Morzy T., Wojciechowski M., Zakrzewicz M., Efficient Constraint-Based Sequential Pattern Mining Using Dataset Filtering Techniques, In: Proceedings of the Baltic Conference, Baltic DB&IS 2002, 1, 213-224, 2002

[24] Orlando S., Perego R., Silvestri C., A new algorithm for gap constrained sequence mining, In: Proceedings of the ACM Symposium on Applied Computing, 540-547, 2004

[25] Pei J. et al., Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach, IEEE Trans. Knowl. Data Eng., 16, 1424-1440, 2004

[26] Pei J. et al., PrefixSpan: Mining sequential patterns by prefix-projected growth, ICDE, 215-224, 2001
[27] Pei J., Han J., Wang W., Constraint-based sequential pattern mining: the pattern-growth methods, J. Intell. Inf. Syst., 28, 133-160, 2007
[28] Pei J., Han J., Wang W., Mining sequential patterns with constraints in large databases, In: Proceedings of the 11th International Conference on Information and Knowledge Management, 18-25, 2002
[29] Qiankun Zhao, Bhowmick S.S., Sequential Pattern Mining: A Survey" Technical Report, CAIS, Nanyang Technological University, Singapore, No. 2003118, 2003
[30] Ren J., Tian H., Shiyong Lv, Sliding-Window Filtering with Constraints of Compactness and Recency in Incremental Database, In: Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management, 2, 665-669, 2008
[31] Srikant R., Agrawal R., Mining quantitative association rules in large relational tables, ACM SIGMOD Record, 25, 1996
[32] Srikant R., Agrawal R., Mining sequential patterns: Generalizations and performance improvements, In: Proceedings of the 5th International Conference on Extending Database Technology (EDBT'96), 3-17, 1996
[33] Tiwari P., Shukla N., Multidimensional Sequential Pattern Mining, Int. J. Sci. Res. Publ., 2, 2012
[34] Zaki M.J., SPADE: An efficient algorithm for mining frequent sequences, Mach. Learn., 42, 31-60, 2001