A Review of “A Multi-Support-Based Sequential Pattern Mining Algorithm” & its optimization Using RBF and Quadratic Programming.

Mr. Hemant Sahu  
HOD of MCA Department  
Geetanjali Institute of Technical Studies, Udaipur  
Rajasthan Technical University, Kota

Dinesh Yadav  
Department of Computer Science & Engineering and MCA  
Geetanjali Institute of Technical Studies, Udaipur  
Rajasthan Technical University, Kota

Abstract: Now a days sequential pattern mining algorithm of data mining is used in various aspect such as biological data analysis, Customer shopping sequences, Medical treatments, natural disasters, science & eng. processes, stocks and markets, Telephone calling patterns, Web access patterns etc. Here we first study the case of M-sequence analysis and its prediction on local support, global support, distribution support for their related respective patterns. As the aforesaid algorithm is the NP-hard type, we also propose its optimization using RBF and Quadratic Programming (QP).

Keywords: Data Mining, Sequential pattern mining, M-sequence analysis, RBF, QP.

1. Introduction  
The major objective of sequential pattern mining is to discover frequent occurring sub-sequences in a large sequence database as patterns. It is an important data mining task with many broad applications like Customer shopping sequences, Medical treatments, Web access patterns and many others. The problem of mining frequent patterns in a set of data sequences was first introduced in [1]. Sequential pattern mining finds frequent sub-sequences whose occurrence frequency in the set of sequences is no less than a user specified min_support threshold [1]. A sequence database stores a number of records, where all records are sequences of ordered events, with or without concrete notions of time. An example sequence database is retail customer transactions or purchase sequences in a grocery store showing, for each customer, the collection of store items they purchased every week for one month. These sequences of customer purchases can be represented as records with a schema [Transaction/Customer ID, <Ordered Sequence Events>], where each sequence event is a set of store items like bread, sugar, tea, milk, and so on. For only two such customers, an example purchase sequential database is [T1, < (bread, milk), (bread, milk, sugar), (milk), (tea, sugar)>]; [T2, < (bread), (sugar, tea)>]. While the first customer, with a transaction ID shown as T1 in the example, made a purchase each of the four weeks, the second customer represented by T2 made purchases during only two of the weeks. Also, a customer can purchase one or more items during each market visit. Thus, records in a sequence database can have different lengths, and each event in a sequence can have one or more items in its set previous studies are obviously we can find that the latter access frequency carried out from two aspects, one is in single sequence[2], the other is in multiple sequences with same type[1,3,4,5,6,7,8,9,10]. Regarding the latter case, the current mining algorithms [1-10] just focus on the case whether the pattern occurs in each sequence, i.e., support indicates the number of sequences, so we just investigate sequential patterns which occur in enough number of sequences, instead of considering the repetitive frequency in each sequence. Correspondingly, the total occurrence frequency in all sequences is not counted in current algorithms Take web access sequences containing 100 users for example, comparing two access pattern: ‘home page news page’ and ‘news page-economy page’. There are 95 users who visit the former and each visit it only once, while just 10 users who visit the latter but each of them 100 times. According to current algorithms, what we may get is that the former occurs 95 times while the latter occurs 10 times for the support means the number of users.
containing pattern. However (1000 times) is larger than the former (95 times). This result is useful for the web designers to optimize the website. Additionally, traditional methods don’t take into account relationship among multiple sequences. Basically, there are three kinds of patterns that should be present in case of multiple sequences mining. The first is the pattern which frequently occurs in one specific sequence, second is the pattern with enough total occurrence frequency in all sequences, and the third one is the pattern which occurs in enough sequences. The first two patterns are not concerned in recent studies. Corresponding to these, we’ll define three types of supports: local support, total support and distribution support in this paper, then we propose a multi-supports-based algorithm for finding above three sequential patterns and its optimization using RBF and Quadratic Programming concepts. According to it we list a new definition for sequence.

Additionally, recent studies classify sequence into two categories: time series [11, 12] is a sequence of data points, measured typically at successive time instants spaced at uniform time intervals. (Like transaction sequence, stock sequence). Time series data have a natural temporal ordering. This makes time series analysis distinct from other common data analysis problems, in which there is no natural ordering of the observations (e.g. explaining people’s wages by reference to their education level, where the individuals’ data could be entered in any order). And spatial sequence [13, 14, 15, 16] where the observations typically relate to geographical locations (e.g. accounting for house prices by the location as well as the intrinsic characteristics of the houses, (such as biological sequences). In fact, time series and spatial sequence are both composed of separate temporal or spatial point and their corresponding value, their sequential pattern mining process is consistent in essence. We introduce a ‘order’ to identify temporal point and spatial point to unify two kinds sequences. The definition and algorithm in this paper is generally available for time series and spatial sequences.

This paper is organized as follows. In section 2, a definition for M-Sequenence is given in section 2.1, three kinds of supports are defined in section 2.2, and three sequential patterns are describes in section 2.3. A multi-supports-based sequential pattern mining algorithm is proposed in section 3. Section 4 discusses the generality of M-Sequences definition, meaning of multiple supports, RBF and Quadratic Programming (QP).and concludes our study in section

2. Multiple supports for sequential pattern mining
2.1 Definition for Sequences
A sequence is an ordered list of objects (or events). Like a set, it contains members (also called elements or terms), and the number of terms (possibly infinite) is called the length of the sequence. For example, <C, R, Y> is a sequence of letters that differs from <Y, C, R> as the ordering matters.

Example1: Given M biological sequences, denoted Example1: Given M biological sequences, denoted as D1, shown in Figure1. Regarding one sub-sequence <ATCCA>, following three cases are available to do sequential pattern mining: (here given M is 20)

{<ACTGT ATCCA GTCTA ACTGT …… AC TGTAC TGCTC ACTGT T> ………………………………
<GGCTA AGTGT ACGGT CGAGA GCAGA … AGTCT TGCTA ACCCT TT>}

Figure 1. Example of biological sequences D1
a. If it occurrence in 5th of sequences is frequent enough (e.g. 25 times). Such sub-sequence is the pattern frequently occurs in one specific sequence of D1. It can reflect the personal feature on one sequence.

b. If it occurrence in every sequence of D1 is (20, 32… 21, 22) times respectively. Then, total occurrence frequency of sub-sequence is 325 times which satisfies a certain number, such sub-sequence is the pattern with enough total occurrence frequency in all sequences. It can reflect features on the whole set.

c. There are 12 sequences containing it in D1. it is the pattern which occurs in enough sequences. When mining the sequential pattern, we should take into consideration of the gap between two elements also. Hereby, we introduce a new attribute ‘order’ in the sequence definition to indicate corresponding temporal point or spatial point of sequence elements.

For example the pattern ‘home page-news page’ is described as < (09:05’30’”, home page), (09:05’45’”, news page)>.

Definition 2.2(Single Sequences): Single sequence is a set of pairs whose form like (order, value). A single sequence S is denoted by <(X1, Y1), (X2, Y2),…, (Xn, Yn)>, where (Xn, Yn) is called an element of sequence. Xn (1<=i<=n) is order, it represents temporal or spatial
information and $X_i < X_j$ $(i<j)$, $Y_l(1<=i<=n)$ is a value of sequence element, especially, Yi can be a value with multiple objects i.e., $(Y_1,Y_2,...,Y_k)$, furthermore, $Y_k$ also would be a complex object.

**Definition 2.3(M Sequences):** M-Sequences is a set of sequences containing $M$ single sequences with same types, it is denoted as:

$$<(X_{11}, Y_{11}), (X_{12}, Y_{12}), ..., (X_{1m}, Y_{1m})>, ...$$

$$<(X_{m1}, Y_{m1}), (X_{m2}, Y_{m2}), ..., (X_{mn}, Y_{mn})>, ...$$

Where $M$ is a positive integer, especially, if $M=1$, it is specified to a single sequence, if $M>1$, it represents multiple sequences. $1<=i<=m$, $n_i$ can be equal or unequal to $n$, $X$ represents the same type order. $Y$ represents the same type value. For brevity, the order ‘$X_{ini}$’ would be omitted if obvious enough in context.

**Definition 2.4(sub-sequence):** A sequence $T = <(x_1, t_1), (x_2, t_2), ..., (x_m, t_m)>$ is a sub-sequence of one sequence S in M-Sequences, $S = <(X_1, Y_{int1}), (X_2, Y_{int2}), ..., (X_m, Y_{intm})>$, where $m<=n$, if $1 <= j <= m - 1, x_j < x_{j+1}$, and $1 <= i <= l, i < j < ... < l$, such that $t_1 \subseteq Y_{int1}, t_2 \subseteq Y_{int2}, ..., t_m \subseteq Y_{intm}$. In this case, $S$ contain $T$, $S$ is also called super-sequence.

If the value $t_j$ $(1<=i<=m)$ of the element $(x_j, t_j)$ in sub-sequence $T$ is equal to the value $Y_{int}(1<=j<=n)$ of the element $(X_{int}, Y_{int})$ in super-sequence $S$, then order $x_j$ should be equal to order $X_{int}$, i.e., $x_j = X_{int}$. Moreover would occur a few times in the super-sequence $S$, so $x_j$ in $T$ have a few related orders $X_{int}$ in $S$.

**Definition 2.5(Sequence element gap):** Sequence element gap is the gap between each two elements in sequence, i.e., the gap between the order $x_{ini}$ and the order $x_{inj}$ $(x_{ini}<x_{inj})$ corresponding to $Y_i$ and $Y_j$, denoted as gap $(x_{ini}, x_{inj}) = x_{inj} - x_{ini} (x_{ini}<x_{inj})$.

**Definition 2.6(Equal gap sub-sequence):** Equal gap sub-sequence is such sub-sequence whose two neighbor elements' sequence element gap, i.e., gap $(x_{ini}, x_{inj})$ is equal when this sub-sequence occurs in super-sequence $T$ every time. Especially, if gap $(x_{ini}, x_{inj}) = 1$, call it as consecutive sub-sequence.

**Definition 2.7(Unequal gap sub-sequence):** If gap $(x_{ini}, x_{inj})$ is unequal when this sub-sequence occurs in super-sequence $T$ every time, call it unequal gap sub-sequence.

**Definition 2.8(Sequential pattern):** Sequential pattern is a sub-sequence whose occurrence frequency satisfies a certain user-specified value.

**Definition 2.9(Sequential pattern mining):** Sequential pattern mining is to find the complete set of sequential patterns in M-sequences.

**2.2. Multiple supports of M-Sequences**

Currently, the support of a sub-sequence $T$ is defined as the number of sequences in the database containing $T$ [1-10]. i.e., if the sub-sequence $T$ occurs in one sequence of database then support is added 1, else unchanged. Under this definition, we can find the patterns that occur in enough sequences. But the method only using one kind of support is partial, we can’t find the patterns that occur frequently in each specific sequence (e.g. one or part users’ access custom in web access), or patterns with enough total occurrence frequency in all sequences (e.g. the pattern ‘A-B’ possesses maximum total access frequency in web access sequences). Here, introduce three supports: local support, total support and distribution support. Given a M-Sequences $D$, sequence $S \subseteq D$, sub-sequence $T (T \subseteq S)$:

**Definition 2.10(Local Support):** sub-sequence occurrence frequency in any one of sequences in $D$ is called local support. Local support of sub-sequence $T$ is the number containing $T$ in $S$, denoted: Localsupp$_D$ $(T)$, i.e.Localsupp$_D$ $(T) = |\{T| T \subseteq S \land S \in D\}|$.

**Definition 2.11(Total Support):** sub-sequence total occurrence frequency in all sequences in $D$ is called total support. Total support of sub-sequence $T$ is the sum of sub-sequence occurrence frequency in each sequence, denoted Totalsupp$_D$ $(T)$, i.e.Totalsupp$_D$ $(T) = \sum |\{T| T \subseteq S \land S \in D\}|$.

**Definition 2.12(Distribution Support):** The number of sequences in M-Sequences containing sub-sequence is called distribution support. Distribution support of sub-sequence $T$ is the number of sequences in $D$ containing $T$, denoted as Distrisupp$_D$ $(T)$, i.e., Distrisupp$_D$ $(T) = |\{S| T \subseteq S \land S \in D\}|$.

Hereby, we use a vector to record distribution support: $(supp,v_1,v_2,...,v_m)$, where $v_i$ $(1<=i<=m)$ indicates whether sub-sequence $T$ occurs in the $i^{th}$ sequence or not, it has two
values: \{0,1\}, ‘supp’ is the value of distribution support, and supp = \kappa \sum_{i=1}^{\text{len}} v_i

Generally, each support has a user-specified min_support threshold. To find various meaningful sequential patterns, the support can be given separately or in combining form. Particularly, when \( M=1 \), M-sequences represents single sequence, only local support is needed.

### 2.3. Sequential pattern mining in M-Sequences

Given an M-Sequences \( D \), there are three definitions for sequential patterns corresponding to three supports:

**Definition 2.13 (Local Sequential Pattern):**
Local sequential pattern is the pattern occurs frequently in each specific sequence in M-sequences. A positive integer \( \zeta_1 \) as the local support threshold, a sub-sequence \( T \) is called a local sequential pattern in M-Sequences if \( \text{Localsupp}_{D}(T) \geq \zeta_1 \), the sub-sequence is local frequent.

**Definition 2.14 (Total Sequential Pattern):**
Total sequential pattern is the pattern with enough total occurrence frequency in all sequences of M-Sequences. A positive integer \( \zeta_2 \) as the total support threshold, a sub-sequence \( T \) is called a total sequential pattern in M-Sequences if \( \text{Totalsupp}_{D}(T) \geq \zeta_2 \), the sub-sequence is total frequent.

**Definition 2.15 (Existence Sequential Pattern):**
Existence sequential pattern is the pattern occurs in enough sequences of M-Sequences. A positive integer \( \zeta_3 \) as the distribution support threshold, a sub-sequence \( T \) is called an existence sequential pattern in M-Sequences if \( \text{Distrisupp}_{D}(T) \geq \zeta_3 \), the sub-sequence is existence frequent.

**Definition 2.16 (Sequential Pattern mining in M-Sequences):**
Sequential pattern mining in M-Sequences is to find local sequential pattern, total sequential pattern and existence sequential pattern.

**Definition 2.17 (Radial basis functions):**
Means to approximate multivariable (also called multivariate) functions by linear combinations of terms based on a single univariate function (the radial basis function).

A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin, so that \( \phi(x) = \phi(||x||) \); or alternatively on the distance from some other point \( c \), called a center, so that \( \phi(x, c) = \phi(||x-c||) \). Any function that satisfies the property \( \phi(x) = \phi(||x||) \) is a radial function.

3. A multi-supports-based sequential mining algorithm

In this section, we will develop a multi-supports based sequential pattern mining algorithm applied in M-Sequences. This algorithm will find not only existence sequential pattern depend on the single support used previously, but also local sequential pattern and total sequential pattern funded on new multiple supports. Data structures and symbol explanation:

The order of the element
The value of sub-sequence elements

![Figure 2. Stack \( t_{\text{len}} \)](image)

D: M-Sequences (see Definition 2.2); S: one sequence of M-Sequences, \( S \in D \); F: A set of frequent sub-sequences whose length are 1, \( F = \{I_1, I_2, \ldots, I_k\} \); \( s_{\text{len}} \): A sub-sequence whose length is \( \text{len} \), the head element of this sub-sequence is \( I_i \) (\( i \in F \)). It is stored in \( t_{\text{len}} \). \( \text{len} \): the length of sub-sequence; \( \text{gap}_{j:j+1} \): gap between the \( j^{\text{th}} \) and \((j+1)^{\text{th}} \) elements in \( s_{\text{len}} \). \( t_{\text{len}} \): A stack corresponding to sub-sequence \( s_{\text{len}} \) (shown in Figure 2). The set of position denoted as \( L \). Each value in \( L.S.p \) indicates the order of each super-sequence \( S \) element corresponding to sub-sequence element. The sum of number in set \( L.S \) is total support, and the sum of number in set \( L \) is total support. It storing position in stack makes reduce searching time complexity, because we take advantage of displacement \( (p_{\text{len}} - p_j) \) to omit irrelevant elements in searching process. \( T_{\text{len}} \): A set of stack \( t_{\text{len}} \).

The general idea of the algorithm: it based on Apriori heuristic [17], if \( s_{\text{len}} \) is not frequent, then we needn’t check \( s_{\text{len+1}} \) i, because occurrence frequency of sub-sequence \( s_{\text{len+1}} \) i exceeding \( s_{\text{len}} \) is impossible.

**The process of this algorithm:**
1. Verify whether \( s_{\text{len}} \) is frequent by checking whether it’s each support be no less than each corresponding support threshold.
2. If \( s_{\text{len}} \) is sequential pattern, then join \( s_{\text{len}} \) with \( F \) to create \( s_{\text{len+1}} \).
3. Check whether $s_{(len+1)}$ i is sequential pattern. If true, then create a stack for it. All these stacks put into a set $T_{(len+1)}$.
4. After sub-sequence in each $t_{m}$ i is joined with F and create new stack, the $T_{m}$ can be deleted.

The overall algorithm is shown in Figure 3.

**Input:** M-Sequences D, local, total, distribution support threshold is $\zeta_{1}$, $\zeta_{2}$, $\zeta_{3}$; sequence element gap.

**Output:** All local, total, existence sequential pattern, and corresponding support.

### 4. Discussion

In our paper, the definition for M-Sequences unifies single sequence and multiple sequences, furthermore, integrates time series and spatial sequences. In practical applications, M-Sequences can be converted to following types: if $M=1$, then M-Sequences represents single sequence; if $M>1$, then it denotes multiple sequences; and that, when the ‘order’ of sequence elements (order, value) is used for temporal point information, M-Sequences indicates time series; when ‘order’ is spatial point, M-Sequences signifies spatial sequences. Virtually, M-Sequences define a kind of sequence data source.

Using three new kinds of supports presented in this paper are available to find more sequential patterns than only a single support earlier. The users can select a support or supports or combining form at will to satisfy more complex application requirements. Example 3: Given a 5-Sequences (M=5) (see table 1):

<table>
<thead>
<tr>
<th>Customer transaction 5-sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Id</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

Using algorithm AprioriAll, only a support given (min_support is 3), we get maximal large sequences: <134>, <135> and <45> [1]. But it can’t show that the purchased volume of item {6} is maximum, because it will not calculate repetitive times of {6}, it only counts twice for two customers contain it. Using total support, the patterns with enough occurrence frequency in all sequences of M-Sequences can be found. And using local support, we can analyze frequent patterns in one specific sequence of M-Sequences. Using local support combined with distribution support, we can analyze whether a few sequences have similar features and generalize them. Using distribution support, patterns which occur in enough sequences of M-Sequences can be found. The multi-supports-based sequential mining algorithm enables to find all new sequential patterns using user-specified support or supports. However, it is just a general algorithm at least and we tried to improve its efficiency with the help of radial basis function and quadratic programming.

### 5. Proposed Work:

As the algorithm here has predefined function on its behalf but we proposed an element first consider on a random basis on the set of frequently used element. Then for RBF we use the function [18].

$$ Y(x) = \sum_{i=1}^{N} W_i \varphi (||X - X_i||) $$

Sum of `N` RBF for its approximation. Now as the function generated on the multiquadratic so the becomes

$$ \Phi (||X - X_i||) = 1 + (\varepsilon (||X - X_i||))^2 $$

So from here we can use the algorithm on the basis of the type taken care, the optimization as the convex constrained problem. We use here projected gradient code described in [20] which just requires us to be able to compute values and gradients of ------ at given pair (f, g) and ------ projection of pairs these in the feasible set. As the algorithm is a vertical type.

$$ \hat{c}_g F (f,g) = \varphi (n(f_i - C_i) + \varphi d) $$

While the projection operator $P_{c}$ is defined by:

$$ C := \{ (f, g) | R_n * R^2 \} $$

$$ P_{c} (f, g) = (f, g') $$

So from using these optimization techniques we can reduce the complexity to $O (n^2)$ to $O (\log n)$ as the prerequisite condition by referring the problem as NP-hard.

In the future we are working on the proposition and its implementation as it is asymptotically data fitting weights go to infinite($P_{i} \rightarrow \infty$) the
problem as taken the shape of
\[
\min_{x} \int_{x} f''(x)^2 \, dx \tag{8}
\]
\[
f(x) = C_i
\]
\[
f(x) \geq 0
\]
From here we also can conclude:
(a) Analytic (closed form) solution.
(b) Smoothness of the reconstructed function.
(c) Non monotone reconstruction from monotone data [19].

6. Conclusions
In this paper, we define the sequence element as a pair (order, value) to unify temporal and spatial sequences. Furthermore, we give a new definition for M-sequences including single sequence and multiple sequences. Specially, three new kinds of supports: local support, total support, and distribution support are introduced. Using these kinds of supports we can realize comprehensively three kinds of sequential pattern mining: local sequential pattern (occurs frequently in each specific sequence of M-Sequences), total sequential pattern (with total occurrence frequency in all sequence of M-Sequences), and existence sequential pattern (which occurs in enough sequences of M-Sequences) according to various applications. Meanwhile, we devise a multi-supports based sequential pattern mining algorithm in relative to sequential pattern mining. As future work, we are going to do more data mining tasks (such as clustering, Outlier detection) for the sequence data source based on multiple support.

6. References

Engineering (ICDE'99), Sydney, Australia, April (1999), pp. 106-115.


