



A two-stage methodology for sequence classification based on sequential pattern mining and optimization

Themis P. Exarchos^{a,b,c}, Markos G. Tsipouras^b, Costas Papaloukas^d, Dimitrios I. Fotiadis^{b,c,*}

^a Department of Medical Physics, Medical School, University of Ioannina, GR 45110 Ioannina, Greece

^b Unit of Medical Technology and Intelligent Information Systems, Department of Computer Science, University of Ioannina, P.O. Box 1186, GR 45110 Ioannina, Greece

^c Institute of Biomedical Technology, CERETETH, GR 41222 Larissa, Greece

^d Department of Biological Applications and Technology, University of Ioannina, GR 45110 Ioannina, Greece

ARTICLE INFO

Article history:

Received 11 February 2008

Received in revised form 16 May 2008

Accepted 24 May 2008

Available online 14 June 2008

Keywords:

Sequential pattern mining
Sequential pattern matching
Sequence classification

ABSTRACT

We present a methodology for sequence classification, which employs sequential pattern mining and optimization, in a two-stage process. In the first stage, a sequence classification model is defined, based on a set of sequential patterns and two sets of weights are introduced, one for the patterns and one for classes. In the second stage, an optimization technique is employed to estimate the weight values and achieve optimal classification accuracy. Extensive evaluation of the methodology is carried out, by varying the number of sequences, the number of patterns and the number of classes and it is compared with similar sequence classification approaches.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Sequence classification is an important problem which arises in many real-world applications, such as protein function prediction, text classification or speech recognition [16]. Sequential data are sequences of ordered “events” representing a situation, where each event might be described by a set of predicates. Examples of sequential data include text, biosequences (DNA, proteins), web-usage data, multiplayer games, plan-execution traces, etc. Classification is the procedure in which given a collection of training records, each one containing a set of attributes and a class, to find a model that maps the features of each record to a class attribute. Subsequently, this model can be used in order to provide predictions for new records. Based on that, given a sequence (constructed from letters drawn from a finite alphabet; i.e. 20-letter alphabet of amino acids in the case of protein classification; a vocabulary of English words in text classification), a sequence classifier assigns a class label (typically drawn from a finite set of mutually exclusive class labels) to this sequence. Data mining and machine learning algorithms offer a number of effective approaches to design sequence classifiers, when a training set of labeled sequences is available [18].

A sequential pattern is a sequence of *itemsets* that frequently occur in a specific order. An *itemset* is a non-empty subset of elements, called *items*, from a set which is called alphabet. In this manner, an *itemset* represents the set of *items* that occur together. Sequential pattern mining is a procedure that discovers sequential patterns existing in databases of sequences. Sequential pattern mining is widely used in a variety of domains, ranging from text to proteins and DNA sequences. The

* Corresponding author. Address: Unit of Medical Technology and Intelligent Information Systems, Department of Computer Science, University of Ioannina, P.O. Box 1186, GR 45110 Ioannina, Greece. Tel.: +30 26510 98803; fax: +30 26510 97092.

E-mail address: fotiadis@cs.uoi.gr (D.I. Fotiadis).

problem was first introduced by Agrawal and Srikant [2], and since then the goal of sequential pattern mining is to discover all frequent sequences of itemsets in a dataset.

The problem of sequence classification has been addressed in the literature in many ways; the earliest approaches employed finite automata and entropy based approaches [29]. Several methodologies have also been proposed which use either hidden Markov models [35,40] or support vector machines [9,27]. Also, several sequence classification methods have been proposed, as applications in specific domains, such as protein classification [11,24,30,36], text classification [20,22], speech [35] and handwriting [19] recognition. A different category of techniques treat the problem of sequence classification as a feature mining problem [25,26,38], i.e. they mine features from a set of training sequences and then use these features as input in a standard classification algorithm. The FeatureMine algorithm uses these features with the naïve Bayes and Winnow algorithm [25,26]. The Classify By Sequences (CBS) algorithm uses a simple scoring function [38]. Tseng and Lee [38] proposed two different approaches, the CBS_ALL and the CBS_CLASS. Experimental results showed that CBS_CLASS outperforms CBS_ALL [38].

Recently, data mining techniques like association rule mining, sequential pattern mining, clustering and classification, emerged in various research topics [1,2,6,34,41]. However, most of the existing data mining methods are designed for solving a specific problem. On the other hand, some few compound methods integrate two or more types of data mining techniques to solve complex problems. These compound methods can effectively utilize the advantages of each individual mining technique to improve the overall performance in data mining tasks. For example, the Classification Based on Associations (CBA) [28] method provides higher accuracy than traditional classification methods such as C4.5 [34]. Hence, it is a promising direction to integrate different types of data mining methods to form a new methodology for solving complex data mining problems.

In this work, we propose a novel methodology for the generation of sequence classification models, that consists of two stages. In the first stage, a sequence classification model based on sequential patterns is created. This first stage is similar to the CBS_CLASS algorithm, which also builds a sequence classification model from the extracted sequential patterns. The innovation of the proposed methodology is the introduction of weights, which are applied to the sequential patterns and to the classes, and their tuning through optimization, during the second stage, which is an extension of the previously reported CBS_CLASS algorithm. The methodology can be considered as a compound data mining method that uses sequential pattern mining for sequence classification. The input to our methodology is a set of labeled training sequences, and the output is a function mapping a new, unknown sequence, to a class. The classification of an unknown sequence is realized automatically. The methodology employs a sequential pattern mining algorithm, a scoring function that uses the sequential patterns for classification and an optimization technique, in order to automatically assign weights to the sequential patterns and to the classes for improving the classification accuracy. The proposed methodology is evaluated using both artificial and real data. Artificial data are employed in order to present a working example of the proposed methodology, while real data correspond to two biological problems of high importance: protein fold recognition and class prediction.

The pattern weights that are assigned to the sequential patterns, after the optimization stage, identify the relative significance of each pattern. The motivation for the use of class weights is that sequential patterns extracted from sequences do not describe all classes with the same adequacy; some classes are over described from the sequential patterns while for others this description is rather poor. Thus, the class weights are introduced to equalize this preference, and subsequently, an optimization technique is used to automatically calculate optimal values for them. To our knowledge, there is no other work in the literature which assigns weights to the extracted sequential patterns and to the classes for sequence classification. Our results indicate that this integration leads to high classification accuracy, superior to previously reported sequence classification methods. Furthermore, the weights assigned to the patterns can provide to the experts additional knowledge on the domain of application, through the identification of the most important patterns. Finally, the proposed methodology is generic and can incorporate different algorithms/approaches in any of its stages.

2. Methods

The list of symbols employed in this work and their explanation are summarized in Table 1. The proposed methodology includes two stages (Fig. 1). In stage 1, a sequence classification methodology is defined, based on sequential patterns. For the realization of stage 1, a dataset $D = \{S_i, c_i\}$, $i = 1, \dots, l_s$, where S_i is a sequence and c_i is its class, with l_c different classes ($c_i = \{1, \dots, l_c\}$) and l_s is the number of sequences in the dataset ($|D| = l_s$), a vector of sequential pattern weights w_p and a vector of class weights w_c are required. This stage is realized in six steps and its outcome is the classification confusion matrix. In stage 2 optimization is performed on an objective function, which is defined by the classification confusion matrix, to find two sets of weights, one for the sequential patterns w_p and one for the classes w_c , which minimize the objective function.

2.1. Stage 1: Sequence classification

Step 1: Sequential pattern mining. The training sequences are divided into l_c subsets, each one containing all sequences belonging to the same class (S^j , $j = 1, \dots, l_c$). Then, sequential pattern mining (SPM) [2] (details on SPM are presented in Appendix A) is applied to each subset, generating l_c sets of sequential patterns (P^j), satisfying the user-defined constraints. The above procedure is also followed by the CBS_CLASS [38] algorithm, and reported to outperform both CBS_ALL, who

Table 1
List of symbols used in the methodology description and their explanation

Symbol	Explanation
$D = \{S_i, c_i\}$	Database of sequences
l_s	Overall number of sequences in D , ($ D $)
S_i	i th sequence
c_i	Class of the i th sequence
l_c	Number of classes
S^j	The subset of sequences belonging to the j th class
SPM	Sequential pattern mining procedure
P^j	j th set of sequential patterns, extracted from S^j
l_j	The number of patterns in P^j
PSM^j	j th pattern score matrix for the P^j
$PSM^j(m, i)$	value of the (m, i) element of the PSM^j matrix, which denotes the score received from the i th sequence if the m th pattern of the j th set of patterns is contained in it
p_m^j	m th pattern of the j th set of patterns
wp^j	Vector of the j th set of pattern weights, $ wp^j = l_j$
wp	Vector of all sets of pattern weights, $ wp = \sum_{j=1}^{l_c} l_j$
CSM	Class score matrix
pc_i	Predicted class of the i th sequence
c_i	Real class of the i th sequence
wc	Vector of class weights, $ wc = l_c$
CM	Confusion matrix
$trace(\cdot)$	The sum of the elements of the main diagonal of a matrix
wp^*	Vector with the optimal j th pattern set weights
wp^*	Vector with the optimal pattern weights of all sets
wc^*	Vector with the optimal class weights
D_{train}	Training set
D_{test}	Test set
I	Different items in the sequences (alphabet)
min_sup	Minimum support (in the SPM algorithm)
max_gap	Maximum gap
min_gap	Minimum gap
$supD(s_a)$	The support of the s_a sequence in the database of sequences D

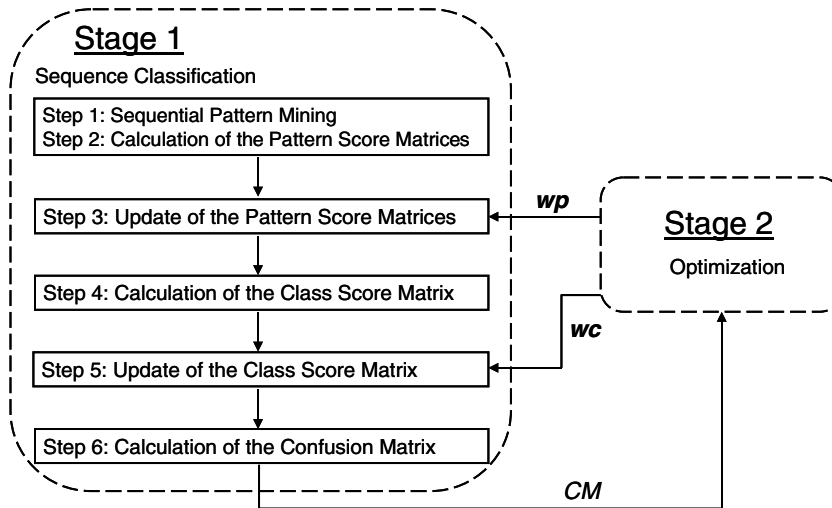


Fig. 1. The two-stage methodology.

mines the whole database of sequences for sequential patterns, and FeatureMine [25,26]. Also this step closely resembles the feature mining problem [25,26]. For this reason, even if SPM is an unsupervised technique, we employ it in a supervised manner, since we generate sequential patterns for each class separately. The output of this stage is the sets of sequential patterns $P^j, j = 1, \dots, l_c$, characterizing the l_c classes.

Step 2: Calculation of the pattern score matrices. After the extraction of the sequential patterns, we create for each class a pattern score matrix $PSM^j, j = 1, \dots, l_c$. Each PSM^j includes a score that defines the implication of a pattern that belongs to class j with all S_i sequences; thus, its size is $l_j \times l_s$. This implication is defined through a scoring function.

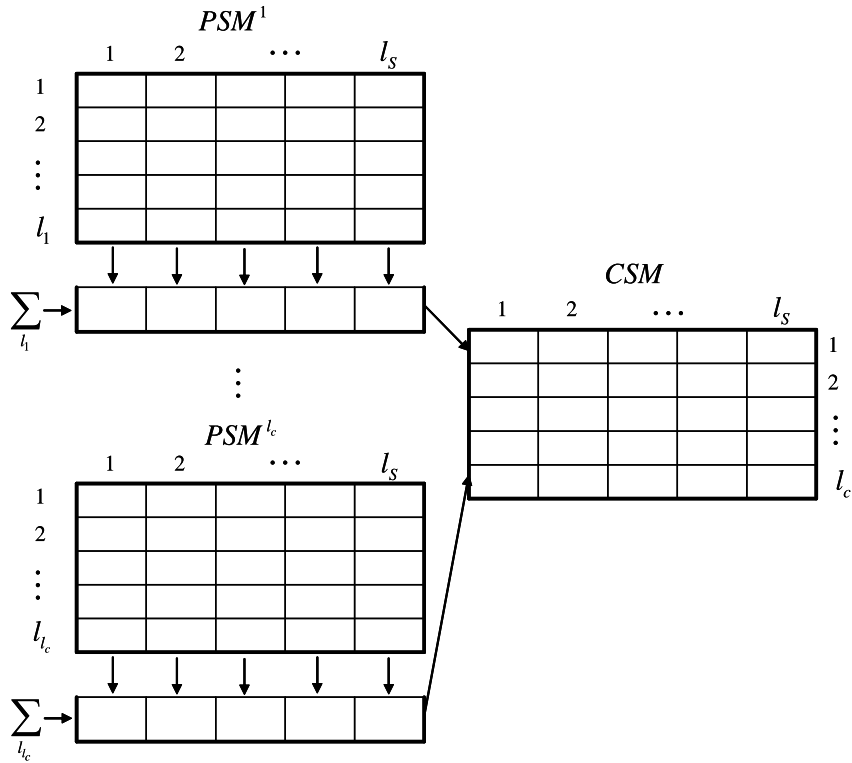


Fig. 2. Calculation of the class score matrix (CSM) from the (updated) pattern score matrix (PSM). The columns of each PSM are summed in order to create the rows of the CSM, which denote the score of each sequence for each one of the classes.

Step 3: Update of the pattern score matrices. Each row of every PSM^j matrix is multiplied by a parameter, which denotes the weight of a specific pattern; thus each matrix PSM^j is updated as follows: $row_j^m = wp^j(m) \cdot row_j^m$, where row_j^m is the m th row of the j th PSM matrix and $wp^j(m)$ is the weight of the m th sequential pattern of the j th pattern set.

Step 4: Calculation of the class score matrix. From the updated PSM^j matrices, we derive the class score matrix (CSM). Each (j, i) element of this matrix is the score of the i th sequence for the j th class, and it is defined as the sum of the scores of all patterns belonging to the j th class for the i th sequence. This is shown schematically in Fig. 2. The size of the CSM matrix is $l_c \times l_s$.

Step 5: Update of the class score matrix. Each row of the matrix CSM is multiplied by a parameter, which denotes the weight of a specific class; thus the matrix CSM is updated as follows: $row_j = wc(j) \cdot row_j$, where row_j is the j th row of the CSM matrix and $wc(j)$ is the j th element of the class weight vector.

Step 6: Calculation of the confusion matrix. For each sequence S_i , a class is predicted (pc_i), based on the updated CSM matrix. The predicted class is determined after estimating the class that obtains the highest score for the i th sequence: $pc_i = \arg \max_{j=1, \dots, l_c} (CSM(j, i))$. Based on the real class c_i and the predicted class pc_i , the confusion matrix (CM) for all the sequences is calculated. The pseudocode for the above six steps of stage 1 is given in Fig. 3.

2.2. Stage 2: Optimization

In this stage, we calculate two sets of weights, wp for the patterns and wc for the classes, which derive the highest classification accuracy of the sequences in the dataset. Initially, a cost (or objective) function is defined:

$$f(D, wp, wc) = g(\text{CM}), \quad (1)$$

where $g(\text{CM})$ is a function that is based on the confusion matrix. Eq. (1) can be formulated as an optimization problem. Different values for the weights have an impact on the cost function g , since they affect CM. The result of the optimization procedure are the optimal weights wp^* , wc^* .

3. Implementation

In order to apply the above described methodology and automatically create a sequence classification model, the following elements need to be defined: (i) the SPM algorithm for the extraction of sequential patterns, (ii) the scoring function for the calculation of the values of all PSM^j matrices and (iii) the optimization elements, such as the objective function of the

STEP 1: Sequential pattern mining (SPM)

- for $j = 1, \dots, l_c$ (for each class)
 - $S^j = \{S_i | c_i = j\}$
(select all sequences from the dataset that belong to the j^{th} class)
 - $P^j = SPM(S^j)$
(perform SPM to these sequences and extract the sequential patterns which describe this class; P^j is the set of these patterns)
- end

STEP 2: Calculation of the pattern score matrices (PSM)

- for $j = 1, \dots, l_c$ (for each class)
 - for $m = 1, \dots, l_j$ (for each pattern of this class, $|P^j| = l_j$)
 - for $i = 1, \dots, l_s$ (for each sequence in the dataset)
Define the value of $PSM^j(m, i)$ based on the scoring function.
 - end
 - end
- end

STEP 3: Update of the pattern score matrices

- for $j = 1, \dots, l_c$ (for each set of sequential patterns P^j)
 - for $m = 1, \dots, l_j$ (for each sequential pattern in every P^j)
 - for $i = 1, \dots, l_s$ (for each sequence in the dataset)
 - $PSM^j(m, i) = wp^j(m) \cdot PSM^j(m, i)$
(Each element of every PSM^j matrix, $(PSM^j(m, i))$, is multiplied with a weight, which denotes the weight of the P_m^j pattern.)
 - end
 - end
 - end
- end

STEP 4: Calculation of the class score matrix (CSM)

- for $j = 1, \dots, l_c$ (for each class)
 - for $i = 1, \dots, l_s$ (for each sequence in the dataset)
 - $CSM(j, i) = \sum_{m=1}^{l_j} PSM^j(m, i)$
(The CSM value for the i^{th} sequence for the j^{th} class is the sum of the scores of all patterns belonging to j^{th} class contained in the i^{th} sequence.)
 - end
 - end

STEP 5: Update of the class score matrix

- for $j = 1, \dots, l_c$ (for each class)
 - for $i = 1, \dots, l_s$ (for each sequence in the dataset)
 - $CSM(j, i) = wc(j) \cdot CSM(j, i)$
(Each element of the columns of the CSM, which denotes the score of the i^{th} sequence for the j^{th} class, is multiplied by a class weight).
 - end
 - end

STEP 6: Calculation of the confusion matrix (CM)

- for $i = 1, \dots, l_s$ (for each sequence in the dataset)
 - $pc_i = \arg \max_{j=1..l_c} (CSM(j, i))$
(pc_i is the predicted class for the i^{th} sequence, defined as the row that exhibits the maximum score in the i^{th} column)
 - $CM(c_i, pc_i) = CM(c_i, pc_i) + 1$
- end

Fig. 3. Pseudocode for the implementation of the six steps of stage 1 of the proposed methodology.

optimization procedure, the optimization algorithm and the optimization approaches for the calculation of the optimal weights.

3.1. Sequential pattern mining algorithm

In the SPM procedure, we can incorporate several constraints, that allow for flexible gap of the extracted sequential patterns [3,8]. Several algorithms have been reported in the literature which implement the SPM procedure [2,4,10,33]. However, little work has been done on constrained SPM [17,37,42]. An algorithm that performs efficient and effective constrained SPM is the cSPADE algorithm [42]. The cSPADE algorithm, which is based on the SPADE algorithm [41], uses efficient lattice search techniques and simple join operations on *id*-lists. As the length of a frequent sequence increases, the size of its *id*-list decreases, resulting in very fast joins. All sequences are discovered with only three database scans, one for frequent 1-sequences, another for frequent 2-sequences, and one more for generating all frequent *k*-sequences. The performance of the cSPADE algorithm has been proven superior, compared to other constrained SPM approaches [17,23,37].

3.2. Scoring function

The scoring function assigns scores to the sequential patterns in the following way: if the P_m^j pattern i.e. the m th pattern of the j th class is contained in the S_i sequence then the (m, i) element of the PSM^j matrix is equal to the length of the P_m^j patterns minus 1, divided by the number of patterns which describe the j th class. If the P_m^j pattern is not contained in the S_i sequence then $PSM^j(m, i)$ is equal to 0. The pseudocode for the implementation of the employed scoring function is shown in Fig. 4.

We subtract 1 from the length of the pattern, in order to assign the minimum score, which is 1, to the minimal pattern, whose length is 2. The length of the pattern in the numerator makes the longer sequential patterns more significant than shorter ones. Also, the score of a sequence with respect to a class is divided by the number of sequential patterns extracted from this class set. Thus, the smaller the number of patterns that describe a class, the more significant is each one of these patterns.

Since the score of a specific pattern for a sequence is 0 if this sequence does not contain the pattern, only the patterns included in a sequence contribute to the class score. It is obvious that the higher the number of large length patterns of a class are contained in a sequence, the higher the score of the sequence for this class is.

3.3. Optimization elements

The objective function that is shown in Eq. (1), takes as input a database of sequences D and the weights w_p and w_c , and the output is a function of the confusion matrix $g(\text{CM})$. The following objective function $g(\text{CM})$ is selected:

$$g(\text{CM}) = l_s - \text{trace}(\text{CM}), \quad (2)$$

since its minimum value is 0, and this value is obtained if (with the appropriate w_p and w_c) all sequences are correctly classified ($l_s = \text{trace}(\text{CM})$). Thus, the minimization of the above objective function, increases the accuracy of the sequence classification model.

The local optimization strategy is chosen, since there is an extremely large number of parameters which must be optimized. In addition, an initial point is available (vectors $w_p = \mathbf{1}$ and $w_c = \mathbf{1}$) and local optimization is significantly faster than global optimization. We have chosen the Roll optimization method [13,32], which is briefly describe in Appendix B. The result of Roll optimization method is a local optimum of the objective function, for w_p and w_c .

The following optimization approaches were employed:

Approach 1 (App. 1): Set both pattern and class weights equal to 1 and calculate the confusion matrix without optimization (i.e. application of stage 1 only of the methodology):

Scoring function

- for $j = 1, \dots, l_c$ (for each class)
 - for $m = 1, \dots, l_j$ (for each pattern of this class, $|P^j| = l_j$)
 - for $i = 1, \dots, l_s$ (for each sequence in the dataset)
 - if P_m^j is contained in S_i then $PSM^j(m, i) = (\text{length}(P_m^j) - 1) / |P^j|$ else $PSM^j(m, i) = 0$.
(if the S_i sequence contains the P_m^j pattern, i.e. the m^{th} pattern of the j^{th} class, then $PSM^j(m, i)$ is equal to $\text{length}(P_m^j) - 1$ divided by the number of patterns describing the j^{th} class, else it is equal to 0).
 - end
 - end
 - end
- end

Fig. 4. Pseudocode of the employed scoring function.

$$f(D, \mathbf{1}, \mathbf{1}) = I_S - \text{trace}(\text{CM}). \tag{3}$$

Approach 2 (App. 2): Set the pattern weights equal to 1 ($wp = \mathbf{1}$) and minimize the objective function to find the optimal class weights wc^* :

$$f(D, \mathbf{1}, wc) = I_S - \text{trace}(\text{CM}). \tag{4}$$

Approach 3 (App. 3): Set the class weights equal to 1 ($wc = \mathbf{1}$) and minimize the objective function to find the optimal pattern weights wp^* :

$$f(D, wp, \mathbf{1}) = I_S - \text{trace}(\text{CM}). \tag{5}$$

Approach 4 (App. 4): From App. 3, set the pattern weights equal to the optimal ones wp^* and minimize the objective function to identify the optimal class weights wc^* :

$$f(D, wp^*, wc) = I_S - \text{trace}(\text{CM}). \tag{6}$$

Approach 5 (App. 5): From App. 2, set the class weights equal to the optimal ones wc^* and minimize the objective function to identify the optimal pattern weights wp^* :

$$f(D, wp, wc^*) = I_S - \text{trace}(\text{CM}). \tag{7}$$

Approach 6 (App. 6): Together optimize both wp and wc , as a single weight vector $[wp, wc]$ and find the optimal $[wp^*, wc^*]$:

$$f(D, wp, wc) = I_S - \text{trace}(\text{CM}). \tag{8}$$

4. Application to artificial data

4.1. Dataset

The artificial dataset is based on the alphabet $I = \{a, b, c\}$, by generating sequences of 4 items, which belong to three classes ($I_c = 3$), thus $D = \{S_i, c_i\}$ with S_i being the sequence and c_i the corresponding class. Twenty four sequences were created, eight from each class. From those, six sequences from each class are used to create the sequence classification model and the remaining are used for testing. Thus, the training dataset D_{train} , consists of 18 sequences ($|D_{\text{train}}| = 18$) and the test dataset D_{test} consists of 6 sequences ($|D_{\text{test}}| = 6$) (Table 2).

4.2. Generation of the sequence classification model

4.2.1. Stage 1: Sequence classification

Step 1: Sequential pattern mining. D_{train} is divided into three subsets, S^1, S^2, S^3 , each one containing sequences from the same class only, thus $|S^1| = 6, |S^2| = 6, |S^3| = 6$. Then, SPM is applied to each set S^j with $min_sup = 50\%$ (i.e. the sequential pattern is present in at least half of the sequences of the respective S^j), $max_gap = min_gap = 1$ (extract only contiguous subsequences as sequential patterns). Based on these, the following sequential patterns are extracted: from the S^1 bb, bc and bbc , from the

Table 2
The artificial dataset

S^j	D_{train}			D_{test}		
	sid	S_i	c_i	sid	S_i	c_i
S^1	1	bbcc	1	1 2	bbbc acc	1 1
	2	baca	1			
	3	abac	1			
	4	bbca	1			
	5	cccb	1			
	6	bbcb	1			
S^2	7	abab	2	3 4	bbba caab	2 2
	8	cbca	2			
	9	cbbb	2			
	10	abaa	2			
	11	cbbc	2			
	12	baba	2			
S^3	13	ccbb	3	5 6	caaa aacc	3 3
	14	bbbb	3			
	15	bcca	3			
	16	acab	3			
	17	acca	3			
	18	acaa	3			

S^2 ab, ba, cb and aba , from the S^3 ac and ca . All patterns have 50% support, except ca that presents 66.7%. Thus, $P^1 = \{bb, bc, bbc\}$, $P^2 = \{ab, ba, cb, aba\}$ and $P^3 = \{ac, ca\}$.

Step 2: Calculation of the pattern score matrices. In order to create the PSM^j matrices, we use the patterns which are contained in each sequence (sequential pattern matching). Here, since the sequential patterns have been extracted using $max_gap = 1$, their matching is performed in the same way, i.e. the sequential patterns exist in the sequences only as contiguous subsequence. Then, for each set P^j and based on the existence or not of a pattern in a sequence, a PSM^j matrix is created, as it is shown in Table 3 for PSM^1 . The values of the entries for all PSM matrices are obtained using the scoring function. For example the first cell of the PSM^1 matrix is 0.33 since the pattern bb is contained in the sequence $bbcc$ and the assigned score is $PSM^1(1,1) = (length(bb) - 1) / |P^1|$, and $length(bb) = 2$, $|P^1| = 3$.

Step 3: Update of the pattern score matrices. Each row of the three PSM matrices is multiplied by a weight $wp^j(i)$, which denotes the weight of the i th pattern of the j th set of pattern (PSM^j). Initially, all $wp^j(i)$ weights are set to 1 so the updated PSM^j matrices remain the same (the weights are tuned in stage 2 of the methodology).

Step 4: Calculation of the class score matrix. From the three PSM matrices, we create the CSM matrix, shown in Table 4, summing the scores of the patterns of the same class (Fig. 3).

The above CSM matrix presented in Table 4, contains for each sequence the score obtained for each class. For example in the first cell, 1.33 is the sum of scores of all patterns of class 1 (P^1) for the $bbcc$ sequence, i.e. $CSM(1,1) = \sum_{m=1}^3 PSM^m(m,1) = 1.33$.

Step 5: Update of the class score matrix. Each row of the CSM matrix is multiplied by a weight $wc(j)$. Initially, all weights are set to 1 so the updated CSM matrix remains the same (the weights are tuned in stage 2 of the methodology).

Step 6: Confusion matrix calculation. Finally, from the updated CSM matrix, we derive the confusion matrix CM for the sequences of the training set D_{train} . Using $pc_i = \arg \max_{j=1, \dots, 3} (CSM(j,i))$, for $i = 1, \dots, 18$, we derive the predicted class (pc_i) by the methodology for each sequence, as it is depicted in Table 5.

From the above, we derive the confusion matrix (CM) for the sequences of D_{train} (Table 6).

The accuracy is given by the trace of the CM , which in our case equals to 10, thus the accuracy is 55.6%.

4.2.2. Stage 2: Optimization

We calculate the optimal values for all $wp^j(m)$ and $wc = [wc(1), wc(2), wc(3)]$ in order to minimize the objective function $f(D_{train}, wp, wc) = |D_{train}| - trace(CM)$, by increasing the $trace(CM)$. The optimization procedure is formulated as: minimize $f(D_{train}, wp, wc)$, subject to wp and wc . As it is shown above, the CM matrix and subsequently the $trace(CM)$ value, are functions of the weights wp and wc . The initial value of the objective function is $f(D_{train}, wp, wc) = 8$. The minimum value of

Table 3
The first pattern score matrix (PSM^1) of the training sequences, from P^1

P^1	$S_i / PSM^1(m,i)$																	
	$bbcc$	$baca$	$abac$	$bbca$	$cccb$	$bbcb$	$abab$	$cbca$	$cbbb$	$abaa$	$cbbc$	$baba$	$ccbb$	$bbbb$	$bcca$	$acab$	$acca$	$acaa$
$P^1 : bb$	0.33	0.00	0.00	0.33	0.00	0.33	0.00	0.00	0.33	0.00	0.33	0.00	0.33	0.33	0.00	0.00	0.00	0.00
$P^2 : bc$	0.33	0.00	0.00	0.33	0.00	0.33	0.00	0.33	0.00	0.33	0.00	0.33	0.00	0.00	0.33	0.00	0.00	0.00
$P^3 : bbc$	0.67	0.00	0.00	0.67	0.00	0.67	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 4
The class score matrix (CSM) of the training sequences

C_j	$S_i / CSM(j,i)$																	
	$bbcc$	$baca$	$abac$	$bbca$	$cccb$	$bbcb$	$abab$	$cbca$	$cbbb$	$abaa$	$cbbc$	$baba$	$ccbb$	$bbbb$	$bcca$	$acab$	$acca$	$acaa$
c_1	1.33	0.00	0.00	1.33	0.00	1.33	0.00	0.33	0.33	0.00	1.33	0.00	0.33	0.33	0.33	0.00	0.00	0.00
c_2	0.00	0.25	1.00	0.00	0.25	0.25	1.00	0.25	0.25	1.00	0.25	1.00	0.25	0.00	0.00	0.25	0.00	0.00
c_3	0.00	1.00	0.50	0.50	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.50	1.00	1.00	1.00	1.00

Table 5
The predicted class pc_i of the S_i sequences

S_i	c_i	pc_i	S_i	c_i	pc_i	S_i	c_i	pc_i
$bbcc$	1	1	$abab$	2	2	$ccbb$	3	1
$baca$	1	3	$cbca$	2	3	$bbbb$	3	1
$abac$	1	2	$cbbb$	2	1	$bcca$	3	3
$bbca$	1	1	$abaa$	2	2	$acab$	3	3
$cccb$	1	2	$cbbc$	2	1	$acca$	3	3
$bbcb$	1	1	$baba$	2	2	$acaa$	3	3

Table 6

The confusion matrix CM of the training sequences

	pc_1	pc_2	pc_3
c_1	3	2	1
c_2	2	3	1
c_3	2	0	4

Table 7

The extracted sequential patterns and their optimal weights

p^1	$wp^{1^*}(m)$	p^2	$wp^{2^*}(m)$	p^3	$wp^{3^*}(m)$
bb	0.93	ab	0.97	ac	0.62
bc	0.94	ba	1.31	ca	0.71
bbc	0.76	cb	1.28		
		aba	0.99		

$f(D, wp, wc)$ is 0, if $\text{trace}(\text{CM}) = |D_{\text{train}}|$, which means that all sequences are classified correctly. We minimize the above objective function using an optimization procedure, with the fourth optimization approach (i.e. two-stage optimization, first optimize $f(D_{\text{train}}, wp, \mathbf{1}) = |D_{\text{train}}| - \text{trace}(\text{CM})$ with respect to wp , so wp^* is calculated, and then optimize $f(D_{\text{train}}, wp^*, wc) = |D_{\text{train}}| - \text{trace}(\text{CM})$ with respect to wc , thus resulting to wc^*). The optimal values wp^* for the extracted sequential patterns are shown in Table 7.

The updated PSM^1 matrix is shown in Table 8, while the new CSM is presented in Table 9.

Based on the updated CSM matrix, we derive the new predictions for each sequence, as it is shown in Table 10.

As we can see in Table 10, the sequence $cbbb$, which was previously classified incorrectly into class 1, is now correctly classified into class 2, with the multiplication of the optimal pattern weights. The new confusion matrix is shown in Table 11.

The new $\text{trace}(\text{CM})$ is 11 and we derive that with the introduction of the optimal weights to the patterns. The accuracy of the method is increased by 5.5% (from 55.6% to 61.1%).

Table 8

The updated pattern score matrix PSM^1 for the training sequences, obtained after multiplying the initial PSM^1 (Table 3) with $wp^{1^*}(m)$

p^1	$S_{ij}/\text{PSM}^1(m, i)$																	
	$bbcc$	$baca$	$abac$	$bbca$	$cccb$	$bbcb$	$abab$	$cbca$	$cbbb$	$abaa$	$cbcb$	$baba$	$ccbb$	$bbbb$	$bcca$	$acab$	$acca$	$acaa$
$P_1^1 : bb$	0.31	0.00	0.00	0.31	0.00	0.31	0.00	0.00	0.31	0.00	0.31	0.00	0.31	0.31	0.00	0.00	0.00	0.00
$P_2^1 : bc$	0.31	0.00	0.00	0.31	0.00	0.31	0.00	0.31	0.00	0.00	0.31	0.00	0.00	0.00	0.31	0.00	0.00	0.00
$P_3^1 : bbc$	0.51	0.00	0.00	0.51	0.00	0.51	0.00	0.00	0.00	0.00	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 9

The updated class score matrix CSM for the training sequences, after the first optimization stage

c_j	$S_j/\text{CSM}(j, i)$																	
	$bbcc$	$baca$	$abac$	$bbca$	$cccb$	$bbcb$	$abab$	$cbca$	$cbbb$	$abaa$	$cbcb$	$baba$	$ccbb$	$bbbb$	$bcca$	$acab$	$acca$	$acaa$
c_1	1.12	0.00	0.00	1.12	0.00	1.12	0.00	0.31	0.31	0.00	1.12	0.00	0.31	0.31	0.31	0.00	0.00	0.00
c_2	0.00	0.33	1.07	0.00	0.32	0.32	1.07	0.32	0.32	1.07	0.32	1.07	0.32	0.00	0.00	0.24	0.00	0.00
c_3	0.00	0.66	0.31	0.35	0.00	0.00	0.00	0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.35	0.66	0.66	0.66

Table 10

The predictions for the training sequences, obtained after the first optimization stage (calculation of the optimal pattern weights)

S_i	c_i	pc_i	S_i	c_i	pc_i	S_i	c_i	pc_i
$bbcc$	1	1	$abab$	2	2	$ccbb$	3	2
$baca$	1	3	$cbca$	2	3	$bbbb$	3	1
$abac$	1	2	$cbbb$	2	2	$bcca$	3	3
$bbca$	1	1	$abaa$	2	2	$acab$	3	3
$cccb$	1	2	$cbcb$	2	1	$acca$	3	3
$bbcb$	1	1	$baba$	2	2	$acaa$	3	3

Table 11

The confusion matrix for the training sequences, obtained after the first optimization stage

	pc_1	pc_2	pc_3
c_1	3	2	1
c_2	1	4	1
c_3	1	1	4

Table 12

The class score matrix of the training sequences, obtained after the second stage of the optimization, multiplying the previous CSM (Table 9), with wc^*

c_j	$S_i/CSM(j,i)$																	
	<i>bbcc</i>	<i>baca</i>	<i>abac</i>	<i>bbca</i>	<i>cccb</i>	<i>bbcb</i>	<i>abab</i>	<i>cbca</i>	<i>cbbb</i>	<i>abaa</i>	<i>cbbc</i>	<i>baba</i>	<i>ccbb</i>	<i>bbbb</i>	<i>bcca</i>	<i>acab</i>	<i>acca</i>	<i>acaa</i>
c_1	1.01	0.00	0.00	1.01	0.00	1.01	0.00	0.28	0.28	0.00	1.01	0.00	0.28	0.28	0.28	0.00	0.00	0.00
c_2	0.00	0.39	1.28	0.00	0.38	0.38	1.28	0.38	0.38	1.28	0.38	1.28	0.38	0.00	0.00	0.29	0.00	0.00
c_3	0.00	0.53	0.25	0.28	0.00	0.00	0.00	0.28	0.00	0.00	0.00	0.00	0.00	0.00	0.28	0.53	0.53	0.53

Table 13

The predictions for the training sequences of the methodology, obtained after the second optimization stage

S_i	c_i	pc_i	S_i	c_i	pc_i	S_i	c_i	pc_i
<i>bbcc</i>	1	1	<i>abab</i>	2	2	<i>ccbb</i>	3	2
<i>baca</i>	1	3	<i>cbca</i>	2	2	<i>bbbb</i>	3	1
<i>abac</i>	1	2	<i>cbbb</i>	2	2	<i>bcca</i>	3	3
<i>bbca</i>	1	1	<i>abaa</i>	2	2	<i>acab</i>	3	3
<i>cccb</i>	1	2	<i>cbbc</i>	2	1	<i>acca</i>	3	3
<i>bbcb</i>	1	1	<i>baba</i>	2	2	<i>acaa</i>	3	3

Having found the optimal pattern weights, we can calculate the optimal class weights. Thus, optimizing the objective function, $f(D_{train}, wp^*, wc) = |D_{train}| - \text{trace}(CM)$, the optimal class weights are identified: $wc^* = [wc^*(1), wc^*(2), wc^*(3)] = [0.9, 1.2, 0.8]$.

With the above class weights, the CSM is updated, as it is shown in Table 12.

Based on the final CSM matrix, we derive new predictions for each sequence (in the case of the sequence *bcca* the score of c_1 is higher than the score of c_3 ; however, the scores have been rounded for simplicity), as shown in Table 13.

As we can see, the sequence *cbca*, which was previously classified incorrectly to class 3, is now correctly classified into class 2, with the multiplication of the optimal class weights. The new confusion matrix is shown in Table 14.

The new trace(CM) is 12 and we derive that with the introduction of the optimal weights for the classes, the accuracy of the method is increased by another 5.6% (from 61.1% to 66.7%). Thus the total increase in accuracy using both sets of weights is 11.1%.

4.3. Evaluation of the generated sequence classification model

The generated sequence classification model was evaluated with six sequences that comprise D_{test} (Table 15).

We test the method using three different approaches:

Table 14

The confusion matrix for the training sequences, obtained after the second optimization stage

	pc_1	pc_2	pc_3
c_1	3	2	1
c_2	1	5	0
c_3	1	1	4

Table 15

The test sequences employed to evaluate the sequence classification model

S_i	c_i	S_i	c_i	S_i	c_i
<i>bbbc</i>	1	<i>bbba</i>	2	<i>caaa</i>	3
<i>accc</i>	1	<i>caab</i>	2	<i>aacc</i>	3

Table 16

The pattern score matrix PSM^1 for the test sequences, obtained from P^1

P^1	$S_i/PSM^1(m,i)$					
	<i>bbbc</i>	<i>acc</i>	<i>bbba</i>	<i>caab</i>	<i>caaa</i>	<i>aacc</i>
P_1^1 : <i>bb</i>	0.33	0.00	0.33	0.00	0.00	0.00
P_2^1 : <i>bc</i>	0.33	0.00	0.00	0.00	0.00	0.00
P_3^1 : <i>bbc</i>	0.67	0.00	0.00	0.00	0.00	0.00

(i) *Set both pattern and class weights to 1*: We create the PSM^j matrices from the P^j sets of patterns, extracted from the model generation. Initially, sequential pattern matching is performed, in the same way as in the model generation (i.e. match the patterns as contiguous subsequences since they have been extracted using $max_gap = 1$). Then, based on the scoring function, the PSM^j matrices for the test sequences are created (e.g. PSM^1 is shown in Table 16). From the above PSM matrices, we derive the CSM matrix, shown in Table 17.

Subsequently the predictions are shown in Table 18. From Table 18, the confusion matrix for the test sequences is derived, shown in Table 19. Thus, $trace(CM) = 3$ and accuracy = 50%.

(ii) *Set pattern weights to the optimal values and the class weights to 1*: If we use the optimal weight for the patterns wp^* obtained from model generation, the updated PSM^j matrices are obtained (e.g. the updated PSM^1 for the test sequences is shown in Table 20). The updated CSM matrix is shown in Table 21. The new predictions are shown in Table 22. From the above, the confusion matrix for the test sequences becomes as shown in Table 23. The new $trace(CM)$ is 4 and we derive that with the introduction of the optimal pattern weights, the accuracy of the method in the test set is increased by 16.7% (from 50% to 66.7%).

Table 17

The class score matrix CSM for the test sequences

C_j	$S_i/CSM(j,i)$					
	<i>bbbc</i>	<i>bccc</i>	<i>bbba</i>	<i>caab</i>	<i>caaa</i>	<i>aacc</i>
C_1	1.33	0.00	0.33	0.00	0.00	0.00
C_2	0.00	0.00	0.25	0.25	0.00	0.00
C_3	0.00	0.50	0.00	0.50	0.50	0.50

Table 18

The obtained predictions for the test sequences

S_i	c_i	pc_i	S_i	c_i	pc_i	S_i	c_i	pc_i
<i>bbbc</i>	1	1	<i>bbba</i>	2	1	<i>caaa</i>	3	3
<i>acc</i>	1	3	<i>caab</i>	2	3	<i>cacc</i>	3	3

Table 19

The confusion matrix for the test sequences

	pc_1	pc_2	pc_3
C_1	1	0	1
C_2	1	0	1
C_3	0	0	2

Table 20

The updated pattern score matrix PSM^1 , obtained by multiplying the initial PSM^1 (Table 16) with wp^{1*}

P^1	$S_i/PSM^1(m,i)$					
	<i>bbbc</i>	<i>acc</i>	<i>bbba</i>	<i>caab</i>	<i>caaa</i>	<i>aacc</i>
P_1^1 : <i>bb</i>	0.31	0.00	0.31	0.00	0.00	0.00
P_2^1 : <i>bc</i>	0.31	0.00	0.00	0.00	0.00	0.00
P_3^1 : <i>bbc</i>	0.51	0.00	0.00	0.00	0.00	0.00

Table 21

The updated class score matrix CSM for the test sequences

c_j	$S_j/CSM(j,i)$					
	<i>bbbc</i>	<i>accc</i>	<i>bbba</i>	<i>caab</i>	<i>caaa</i>	<i>aacc</i>
c_1	1.12	0.00	0.31	0.00	0.00	0.00
c_2	0.00	0.00	0.33	0.24	0.00	0.00
c_3	0.00	0.31	0.00	0.35	0.35	0.31

Table 22

The predictions of the methodology, using the optimal pattern weights, for the test sequences

S_i	c_i	pc_i	S_i	c_i	pc_i	S_i	c_i	pc_i
<i>bbbc</i>	1	1	<i>bbba</i>	2	2	<i>caaa</i>	3	3
<i>accc</i>	1	3	<i>caab</i>	2	3	<i>cacc</i>	3	3

Table 23

The confusion matrix, using the optimal pattern weights, for the test sequences

	pc_1	pc_2	pc_3
c_1	1	0	1
c_2	0	1	1
c_3	0	0	2

Table 24

The final class score matrix CSM of the test sequences

c_j	$S_j/CSM(j,i)$					
	<i>bbbc</i>	<i>accc</i>	<i>bbba</i>	<i>caab</i>	<i>caaa</i>	<i>aacc</i>
c_1	1.01	0.00	0.28	0.00	0.00	0.00
c_2	0.00	0.00	0.39	0.29	0.00	0.00
c_3	0.00	0.25	0.00	0.28	0.28	0.25

Table 25

The final predictions of the methodology for the test sequences

S_i	c_i	pc_i	S_i	c_i	pc_i	S_i	c_i	pc_i
<i>bbbc</i>	1	1	<i>bbba</i>	2	2	<i>caaa</i>	3	3
<i>accc</i>	1	3	<i>caab</i>	2	2	<i>cacc</i>	3	3

Table 26

The final confusion matrix of the test sequences

	pc_1	pc_2	pc_3
c_1	1	0	1
c_2	0	2	0
c_3	0	0	2

(iii) *Set pattern and class weights to the optimal values:* If we introduce also the optimal class weights, the final CSM matrix is shown in Table 24. The final predictions are presented in Table 25. From the above predictions, the final confusion matrix for the test sequences is extracted and it is shown in Table 26. The new trace(CM) is 5 so with the introduction of the optimal weights to the classes, the accuracy of the method is increased by 16.7%. Thus, the final classification accuracy using both sets of weights is increased up to 83.3%.

5. Application to real data

The proposed methodology is evaluated using a biological sequence dataset. Results of the proposed methodology are presented without the use of stage 2 [14,15] and with the use of stage 2, by applying all five different optimization approaches (App. 2–App. 6).

5.1. Dataset

The sequence classification domain which is selected is the classification of protein primary structures into folds and classes. The formulation of SPM covers almost any categorical sequential domain [39]. In order to apply SPM to a specific domain, the following notions are required: a database of sequences D , a set of items (alphabet) I , a definition of the transaction id (tid) and a definition of an itemset. In what concerns our problem, the database D consists of protein primary structures and each one of them has a sequence id . The set of items I is the 20 amino acids that compose the protein primary structures plus one for the unknown amino acid. An itemset in a transaction consists of a single item (one of the 21 letters) while the tid is the position of the amino acid in the protein primary structure, rather than the time.

A group of primary protein sequences is taken from the Protein Data Bank (PDB) [7]. All members of this group correspond to a specific fold of the structural classification of proteins (SCOP) database [31]. Specifically the 17 SCOP folds, with at least 30 members, from classes A and B are used to derive the training and test data. From the resulted 1000 proteins, the two thirds from each category are randomly selected and used for training, while the rest for testing.

From the above sequences, four datasets, each one with different classification complexity, are derived and four different classification experiments are performed. Each dataset is divided into training and test sets:

- In the first experiment (Exp. 1), we use sequences from 17 categories (class A and class B folds). The training and test sets consist of 666 and 334 proteins, respectively.
- In the second experiment (Exp. 2), we use sequences from 10 categories (class B folds). The training set consists of 406 proteins and the test set of 203 proteins.
- In the third experiment (Exp. 3), we use sequences from seven categories (class A folds). The training set consists of 260 proteins and the test set of 131 proteins.
- In the fourth experiment (Exp. 4), we use sequences from two categories (All sequences from class A folds were considered to belong to the first category and all sequences from class B folds to the second category). Six hundred and sixty proteins constitute the training set and 334 proteins the test set.

5.2. Sequence classification model generation

In all experiments, we set the minimum support to 50%, i.e. a sequential pattern is frequent if it is contained in at least half of the sequences of a specific class. Also, for each of the above experiments, we vary the number of max_gap from 1 to 3, thus creating in total 12 experiments, since the classification is performed using the extracted sequential patterns. It should be mentioned that in the above experiments, the training set is used for sequential pattern mining, as well as, for the calculation of wp^* and wc^* . (Stage 1 of the methodology is not described here in detail due to the large number of the extracted sequential patterns.)

5.3. Evaluation of the generated sequence classification model – comparative study

In the evaluation phase, the sequential patterns and the weights wp^* , wc^* calculated during the sequence classification model generation, are used to classify the sequences of the test set. In addition, we employed the CBS algorithm and the FeatureMine algorithm for comparison. These algorithms perform sequence classification using sequential patterns. Both algorithms are tested using the same experimental procedure. We implemented the CBS_CLASS variation of the CBS algorithm. In the CBS_CLASS algorithm the training set is divided into subsets of sequences belonging to the same category. These subsets are mined using sequential pattern mining and a set of sequential patterns is derived for each class. Then, the score of every sequential pattern is calculated for each sequence, as follows: if the P_m^j pattern, is contained in the S_i sequence then the (m, i) element of the PSM^j matrix is equal to $length(P_m^j) / \sum_{m=1}^j length(P_m^j)$, else it is equal to 0. After the creation of the PSM matrix, the CBS_CLASS algorithm employs the same stages for the creation of the CSM matrix and the classification of a sequence in a predicted class (pc) with the proposed methodology, but without the use of the optimization stage.

The FeatureMine algorithm mines also the training sequence database for class-specific sequential patterns. The extracted patterns are then used to create a Boolean matrix. Every extracted pattern is considered as a feature, and if a pattern is contained in a sequence, the Boolean matrix is filled with 1, else it is filled with 0, in the corresponding position. The Boolean matrix is the transformed dataset and it is further used by standard classification algorithms such as Naïve Bayes and Winnow. In order to use sequential patterns as features, these should be frequent, distinctive of at least one class and the corresponding sets should not contain any redundant features.

Table 27

Number of features used, optimization parameters and reported accuracy (Acc. %) in the training and test sets for the first experiment, for all values of *max_gap*, for the CBS and the FeatureMine algorithms and the six different optimization approaches of the proposed methodology

Exp. 1: $ D_{\text{train}} = 666$, $ D_{\text{test}} = 334$ and $l_c = 17$								
	Comparative study		Proposed methodology					
	CBS	FeatureMine	App. 1	App. 2	App. 3	App. 4	App. 5	App. 6
<i># feat</i>								
Gap 1	1568	107	1568	1568	1568	1568	1568	1568
Gap 2	3670	137	3670	3670	3670	3670	3670	3670
Gap 3	7404	188	7404	7404	7404	7404	7404	7404
<i>#param</i>								
Gap 1	–	–	–	17	1568	1568 + 17	17 + 1568	1585
Gap 2	–	–	–	17	3670	3670 + 17	17 + 3670	3687
Gap 3	–	–	–	17	7404	7404 + 17	17 + 7404	7421
<i>training acc (%)</i>								
Gap 1	36.5	59.8	36.5	58.7	52.3	54.4	63.8	57.5
Gap 2	32.0	61.1	38.4	62.3	57.5	58.7	67.3	64.1
Gap 3	22.7	71.0	54.4	65.5	63.7	66.2	67.7	67.9
<i>test acc (%)</i>								
Gap 1	22.2	35.0	22.5	31.4	26.7	27.5	35.6	31.7
Gap 2	19.2	32.6	18.3	32.9	28.7	28.4	32.6	33.2
Gap 3	14.1	41.0	27.0	36.5	33.5	36.2	35.3	36.5

Accuracy results depicted with bold denote the highest accuracy obtained with the respective gap for a specific experiment.

Table 28

Number of features used, optimization parameters and reported accuracy (Acc. %) in the training and test sets for the second experiment, for all values of *max_gap*, for the CBS and the FeatureMine algorithms and the six different optimization approaches of the proposed methodology

Exp. 2: $ D_{\text{train}} = 406$, $ D_{\text{test}} = 203$ and $l_c = 10$								
	Comparative study		Proposed methodology					
	CBS	FeatureMine	App. 1	App. 2	App. 3	App. 4	App. 5	App. 6
<i># feat</i>								
Gap 1	1142	95	1142	1142	1142	1142	1142	1142
Gap 2	2444	118	2444	2444	2444	2444	2444	2444
Gap 3	5035	181	5035	5035	5035	5035	5035	5035
<i>#param</i>								
Gap 1	–	–	–	10	1142	1142 + 10	10 + 1142	1152
Gap 2	–	–	–	10	2444	2444 + 10	10 + 2444	2454
Gap 3	–	–	–	10	5035	5035 + 10	10 + 5035	5045
<i>training acc (%)</i>								
Gap 1	43.8	64.0	44.1	66.5	65.8	69.5	72.9	70.2
Gap 2	28.1	65.8	34.2	63.1	59.9	64.8	69.7	67.7
Gap 3	15.3	74.6	61.6	71.2	71.4	72.4	72.9	72.4
<i>test acc (%)</i>								
Gap 1	28.1	41.4	30.1	38.4	40.9	41.4	36.0	40.4
Gap 2	16.8	40.9	18.2	37.0	35.5	36.0	37.4	35.0
Gap 3	12.8	41.9	33.5	43.8	41.4	40.9	44.8	43.8

Accuracy results depicted with bold denote the highest accuracy obtained with the respective gap for a specific experiment.

Tables 27–30 present the number of features used, the optimization parameters and the reported accuracy in the training and test sets, for all experiments, all values of *max_gap*, for the CBS algorithm and FeatureMine algorithms and the six different approaches of the proposed methodology.

6. Discussion

We presented a novel methodology for the automated generation of sequence classification models, that can be applied in any (discrete) sequential domain. Initially, sequential patterns are extracted from a set of (training) sequences. The scores for each sequential pattern and each class are computed. In addition, optimal weights for each pattern and for each class are calculated using an optimization technique. The obtained optimal pattern and class weights along with the extracted sequential patterns compose the sequence classification model, which is used to classify the test sequences.

Table 29

Number of features used, optimization parameters and reported accuracy (Acc. %) in the training and test sets for the third experiment, for all values of max_gap , for the CBS and the FeatureMine algorithms and the six different optimization approaches of the proposed methodology

Exp. 3: $ D_{train} = 260$, $ D_{test} = 131$ and $l_c = 7$								
	Comparative study		Proposed methodology					
	CBS	FeatureMine	App. 1	App. 2	App. 3	App. 4	App. 5	App. 6
<i># feat</i>								
Gap 1	426	57	426	426	426	426	426	426
Gap 2	1226	72	1226	1226	1226	1226	1226	1226
Gap 3	2369	83	2369	2369	2369	2369	2369	2369
<i>#param</i>								
Gap 1	–	–	–	7	426	426 + 7	7 + 426	433
Gap 2	–	–	–	7	1226	1226 + 7	7 + 1226	1233
Gap 3	–	–	–	7	2369	2369 + 7	7 + 2369	2376
<i>training acc (%)</i>								
Gap 1	59.2	66.2	59.2	66.9	68.1	70.0	71.5	70.8
Gap 2	61.2	70.8	67.3	78.1	83.1	83.1	80.0	83.1
Gap 3	65.4	78.5	61.9	77.7	76.2	78.9	80.0	77.7
<i>test acc (%)</i>								
Gap 1	42.8	48.1	42.8	42.0	42.0	40.5	46.6	40.5
Gap 2	37.4	45.8	33.6	45.8	40.5	40.5	47.3	42.0
Gap 3	50.4	51.9	42.0	52.7	51.9	51.9	51.9	57.3

Accuracy results depicted with bold denote the highest accuracy obtained with the respective gap for a specific experiment.

Table 30

Number of features used, optimization parameters and reported accuracy (Acc. %) in the training and test sets for the fourth experiment, for all values of max_gap , for the CBS and the FeatureMine algorithms and the six different optimization approaches of the proposed methodology

Exp. 4: $ D_{train} = 666$, $ D_{test} = 334$ and $l_c = 2$								
	Comparative study		Proposed methodology					
	CBS	FeatureMine	App. 1	App. 2	App. 3	App. 4	App. 5	App. 6
<i># feat</i>								
Gap 1	1568	47	1568	1568	1568	1568	1568	1568
Gap 2	3670	74	3670	3670	3670	3670	3670	3670
Gap 3	7404	127	7404	7404	7404	7404	7404	7404
<i>#param</i>								
Gap 1	–	–	–	2	1568	1568 + 2	2 + 1568	1570
Gap 2	–	–	–	2	3670	3670 + 2	2 + 3670	3672
Gap 3	–	–	–	2	7404	7404 + 2	2 + 7404	7406
<i>training acc (%)</i>								
Gap 1	61.4	76.0	62.3	82.1	83.6	83.6	84.2	83.6
Gap 2	51.2	80.3	53.0	85.3	85.9	85.9	86.5	84.7
Gap 3	49.1	82.3	66.8	84.4	80.2	81.7	84.4	84.2
<i>test acc (%)</i>								
Gap 1	59.6	68.9	60.2	76.1	75.5	75.5	75.5	75.5
Gap 2	49.4	69.8	51.2	74.9	77.0	77.0	74.6	73.7
Gap 3	48.5	72.2	62.0	77.8	75.5	75.8	77.8	77.6

Accuracy results depicted with bold denote the highest accuracy obtained with the respective gap for a specific experiment.

The proposed methodology introduces several innovative features. To our knowledge, the automatic assignment of weights to patterns and classes and their tuning using optimization techniques, for classification purposes is proposed for the first time in the literature. Other similar approaches use the extracted sequential patterns either as input features [25,26] to standard classification algorithms, or employ a scoring function, similar to the one reported in the current work [14,15,38]. The weight assignment to the patterns and to the classes and their tuning through optimization, is a major advantage of our methodology, since it adjusts the descriptive ability of the set of patterns for each class, thus leading to high classification accuracy, better than that of previous works. Also, the optimal weights of the patterns can provide to the domain experts new knowledge related to the significance of each pattern, like for example the case of biologically significant protein patterns.

The presented two-stage procedure is generic and different components can be used for any stage of the methodology, i.e. different SPM algorithm and/or scoring function and also alternative objective function and/or optimization method (local or global). The SPM approach, employed in this work, is suitable for analyzing sequences and it is able to discover strong

sequential dependencies (patterns). In addition, the use of sequential pattern mining leads to pattern discovery in the specific sequential domain of application. Furthermore, the training phase of the method, i.e. the determination of the sequential patterns, is a fast procedure due to the use of the cSPADE algorithm. In general, SPM is a time consuming process and requires high computational load which is increased exponentially since longer sequences need to be mined. The lattice search techniques and the simple joins, that the cSPADE algorithm employs, handle the two above aspects effectively. It should be mentioned that the employed scoring function is selected heuristically, obtained after a series of experiments. For example, we utilized also as scoring function the times a sequential pattern is contained in the sequence raised in the power of n ($n = 1, 2, \dots$), the logarithm of the length of the pattern, the length of the pattern raised in the power of n ($n = 1, 2, \dots$), the support of the pattern and others. All the above reported lower classification results when they were used in our sequence database.

The proposed methodology has been evaluated systematically, using 12 different evaluation experiments (four datasets multiplied by three different values of *max_gap*). In the design of the classification experiments, special attention was given to create classification experiments with different properties and classification difficulty; the length of the employed sequences ranges from 36 to 590 letters, using a 21 letter alphabet, while the number of classes, is 17, 10, 7 and 2. Also the number of sequential patterns extends from 426 to 7404. In addition, stage 2 of the methodology is implemented with five different optimization approaches (plus one without the use of the optimization stage). This large number of different evaluation experiments, resulting from the wide range of parameters, ensures the reliable evaluation of the proposed methodology.

FeatureMine presents the lowest computational complexity compared to all other methodologies, due to the employment of the reduced set of sequential patterns. App. 1 and CBS have the same computational complexity both in training and testing, since for each value of *max_gap*, the same number of sequential patterns is both mined in training and matched in testing. Approaches 2–6 present higher computational complexity in the training phase, due to the fact that they employ two stages, the first stage which is common with App. 1 and CBS and the second stage, which employs an optimization technique. Thus, the additional complexity of the Approaches 2–6 is due to the optimization stage and depend on the selection of the optimization technique. In the current work, the Roll method is employed, which is a local optimization technique with low computational complexity. This complexity depends on the number of classes and the number of sequential patterns, since their value defines the number of parameters for optimization. App. 2 presents lower computational complexity compared to Approaches 3–6, since in App. 2 only the class weights are optimized, and not the pattern weights. It should be mentioned that the number of extracted sequential patterns highly depends on the value of *max_gap*, thus as *max_gap* increases the number of sequential patterns and subsequently the computation cost for all methodologies increases.

The proposed novel introduction of weights and their optimization, significantly improves the ability of the sequential patterns to classify sequences, by adjusting the relative importance of each class according to the obtained optimal weights. More specifically, in all 12 classification experiments (Tables 27–30), both in training and testing, the proposed methodology (App. 2–App. 6) presents significantly higher accuracy than the approach that uses only stage 1 (App. 1); the average accuracy increase between App. 1 and the best result obtained from App. 2 to App. 6 is 22% and 14.1%, for the training and test set, respectively. Thus, the weight introduction and their optimization has a major impact on the classification accuracy of the sequence classification model.

A comparison between the five different approaches of the proposed methodology that employ the weight introduction and optimization (App. 2–App. 6) shows that there is single approach that performs better in all different experiments. However, App. 5 is slightly better overall, compared to the other approaches, but not in every different experiment.

App. 5 is an extension of App. 2, which is in turn an extension of App. 1. In App. 1, both sets of weights are set equal to 1, since this is the initial value, considered when only stage 1 of the methodology is used. Subsequently, App. 2 extends App. 1 by introducing an optimization stage for the class weights (*wc*), thus adapting the relative importance of the sets of the sequential patterns to the optimal value (*wc**). App. 5 uses the optimal class weights, produced by App. 2 and further opti-

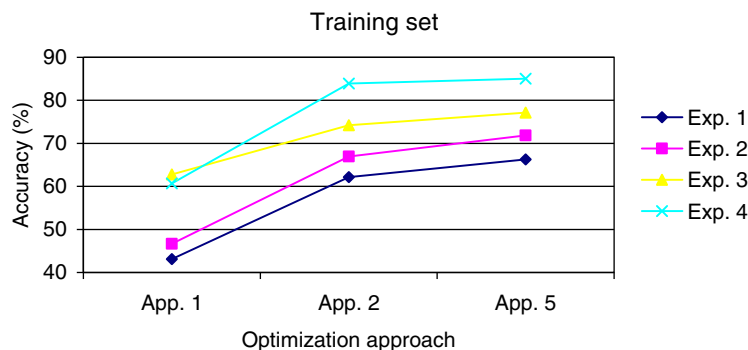


Fig. 5. Variation of the accuracy from App. 1 (no optimization) to App. 2 (class weight optimization) and to App. 5 (class weight and then pattern weight optimization) in the training set.

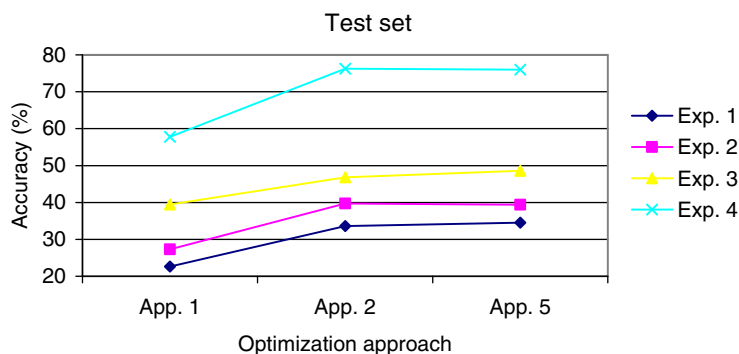


Fig. 6. Variation of the accuracy from App. 1 (no optimization) to App. 2 (class weight optimization) and to App. 5 (class weight and then pattern weight optimization) in the training set.

mizes the model by calculating optimal values for the pattern weights (wp), i.e. identifying the relative importance of each extracted sequential pattern (wp^*). This gradual improvement of the model, resulting from the increase of the introduced parameters (weights) and their optimization is also reflected in the obtained results. Figs. 5 and 6 present the average results in each of the four different experiments, for the training and test sets, respectively. The same strategy is followed by App. 1 (no optimization) to App. 3 (optimization of the pattern weights) and then to App. 4 (optimization of the pattern and then of the class weights), and the respective results follow also the same increasing trend.

It should be mentioned that there is a considerable difference between the classification accuracy in the training and test sets. This is mainly attributed to the increased classification difficulty of the problem and less to overfitting phenomena, since, in most cases, increment of the training accuracy results to higher classification ability in the test set. The difference in accuracy between the simple sequential pattern based classifier (App. 1) and the other approaches (App. 2–6) is statistically significant, at 95% confidence level. However, the extension of App. 2–App. 5 does not present a statistically significant difference in the classification accuracy. The same applies to App. 3 and its extension, App. 4.

A comparative study has been conducted between the different approaches of the proposed methodology and the CBS and FeatureMine algorithms. The average accuracy of each different experiment, for the CBS algorithm, the FeatureMine algorithm and the six different approaches of the proposed methodology, are presented in Figs. 7 and 8, for the training set and the test set, respectively. The best average accuracy in the training set for all four experiments, is reported by App. 5. In the test set, in Exp. 1 and 2, FeatureMine presents higher average accuracy, in Exp. 3, the proposed methodology and FeatureMine present similar average accuracy, and in Exp. 4, the proposed methodology outperforms FeatureMine.

The scoring function proposed in this work (mentioned as App. 1 in Tables 27–30), presents superior classification accuracy than the one employed in the CBS algorithm, as it is shown in Tables 27–30. More specifically, the accuracy obtained in almost all classification problems (in 11 out of 12 experiments) is improved during training, when the proposed scoring function is employed, instead of the one used in the CBS algorithm. This improvement also holds in the testing (in 9 out of 12 experiments).

Comparing directly the proposed methodology with the FeatureMine algorithm, the proposed methodology presents better results than FeatureMine in 9 out of the 12 classification experiments in the training set and in 8 out of the 12 classification experiments in the test set (plus one where both the proposed methodology and FeatureMine have the same

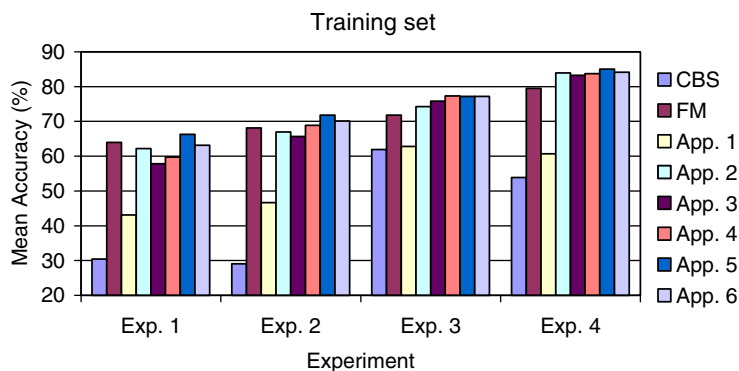


Fig. 7. Graphical representation of the average accuracy in terms of each different experiment, for the CBS algorithm, the FeatureMine algorithm and the six different approaches of the proposed methodology, in the training set.

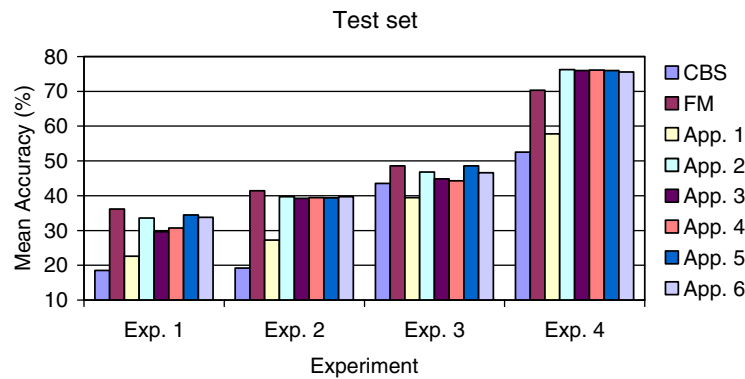


Fig. 8. Graphical representation of the average accuracy in terms of each different experiment, for the CBS algorithm, the FeatureMine algorithm and the six different approaches of the proposed methodology, in the test set.

accuracy). The advantage of FeatureMine compared to the proposed methodology is the lower computational complexity as it is shown in Tables 27–30 (number of features and optimization parameters); however, our method provides the experts with additional information related to the domain of application, since the pattern and class weights define the relative significance of each pattern and class, respectively. A direct comparison with CBS shows that the proposed methodology outperforms CBS in all 12 classification experiments, both in training and test sets while most of the times presents much higher classification accuracy.

Since the methodology presented here is based on sequential pattern mining, inevitably suffers from the large number of discovered patterns which increases exponentially with *max_gap*. Although, the extraction of sequential patterns is relatively fast (due to the cSPADE algorithm), the overall processing time remains high. In addition, SPM, besides discovering valid and causal relationships in the sequential data, will also find spurious and particular relationships among the data in the specific dataset. The above issues could be treated by employing a pattern reduction/selection algorithm, something that we intend to work with in the future. The optimization stage, although it significantly improves the classification accuracy of the current approach increases even more the computational effort and the overall time for training. For this reason, a local optimization strategy was selected, which however, does not ensure the best results (global optimization could be feasible in case where the number of patterns is reduced).

7. Conclusions

A two-stage methodology for sequence classification has been presented along with an extensive evaluation. The methodology provides high classification results in the sequence classification problem, comparable or better with previously reported works. The optimization stage introduced significantly improves the results and the optimal calculated parameters can provide significant knowledge to the experts of the domain of application. Future work will focus on the use of methods for sequential pattern selection and the employment of specific types of patterns such emergent sequential patterns [12], minimal distinguishing sequential patterns [21], or sequential patterns extracted using contrast data mining techniques [5]. Also, the employment of different scoring functions and other machine learning approaches, such as lineal modelling or neural networks, for identifying the optimal weight values will be addressed. Finally, the extension of the methodology in order to handle time series, through the use of discretization techniques will be examined.

Appendix A. Sequential pattern mining – definitions and terminology

Sequential pattern mining is a common form of local-pattern discovery in unsupervised learning systems. The problem of SPM is defined as follows [2]: Let $I = \{i_1, i_2, \dots, i_o\}$ be a set of items. A subset $X \subseteq I$ is an itemset and $|X|$ is the size of X . A sequence $s = (s_1, s_2, \dots, s_p)$ is an ordered list of itemsets, where $s_i \subseteq I$, $i \in \{1, \dots, p\}$. The size, p , of a sequence is the number of itemsets in the sequence, i.e. $|s|$. The length l of a sequence $s = (s_1, s_2, \dots, s_p)$ is defined as $\sum_{i=1}^p |s_i|$. A sequence with length l is an l -sequence. A sequence $s_a = (a_1, a_2, \dots, a_q)$ is contained in another sequence $s_b = (b_1, b_2, \dots, b_r)$ if there exist integers $1 \leq i_1 < i_2 < \dots < i_q \leq r$ such that $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \dots, a_q \subseteq b_{i_q}$.

In SPM, a database D is a set of tuples (sid, tid, X) , where *sid* is a sequence-id, *tid* is a transaction-id based on the transaction time and X is an itemset such that $X \subseteq I$. Each tuple in D is referred to as a transaction. For a given sequence-id, there are no transactions with the same *tid*. All transactions with the same *sid* can be viewed as a sequence of itemsets ordered by increasing *tid*. Thus, an analogous representation for the database is a set of sequences of transactions and we refer to this dual representation of D as its sequence representation.

The support of a sequence s_a in the sequence representation of a database D is defined as the percentage of sequences $s \in D$ which contain s_a and is denoted by $supD(s_a)$. Given a support threshold *minSup*, a sequence s_a is a frequent l -sequential

pattern on D (or frequent l -sequence) with l being the length of the sequence, if $\text{sup}D(s_a) \geq \text{minSup}$. The problem of mining sequential patterns is to find all frequent sequential patterns for a database D , given a support threshold minSup .

A.1. Sequential pattern mining constraints

Several constraints can be incorporated when mining sequential patterns [3,8]. One of the simplest constraints applied is the gap constraint. This constraint imposes a limit in the maximum distance between two consecutive itemsets in the sequence. This simple constraint can be used to reflect the impact of an item on another one, in particular, when each transaction occurs at a particular instant of time (position). When using gap constraints, the notion of “contained in” is adapted: a sequence $s_a = (a_1, a_2, \dots, a_q)$ is a δ -distance subsequence of $s_b = (b_1, b_2, \dots, b_r)$, if there exist integers $1 \leq i_1 < i_2 < \dots < i_n \leq r$ such that $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \dots, a_q \subseteq b_{i_q}$ and $i_s - i_{s-1} \leq \delta$. A sequence s_a is a contiguous subsequence of s_b if s_a is a 1-distance subsequence of s_b , i.e. the items of s_a can be mapped to a contiguous segment of s_b . Using $\delta = 1$ (maximum gap = 1) the possibility of having gaps between consecutive items is eliminated. Similar to the maximum gap constraint is the minimum gap constraint, which states that the distance between two consecutive items must be larger than a specified value ($i_s - i_{s-1} > \delta'$).

Appendix B. Roll optimization method

The Roll optimization method [13,32] works as follows: for each variable X_i there exists an associated step s_i . For a problem where the objective function depends on n variables the procedure described below is repeated n times, $i = 1, 2, 3, \dots, n$. Let $X^c = (X_1^c, X_2^c, \dots, X_n^c)$ be the current point and let $f_c = f(X^c)$.

- (1) Pick a trial point: $X_j^t = X_j^c$ for all $j \neq i$ and $X_i^t = X_i^c + s_i$.
- (2) Calculate $f_+ = f(X^t)$.
- (3) If $f_+ < f_c$ set $X^c = X^t$, $f_c = f_+$ and $s_i = as_i$ proceed from step 1 for the next value of i .
- (4) If $f_+ \geq f_c$ pick another trial point as: $X_j^t = X_j^c$ for all $j \neq i$ and $X_i^t = X_i^c - s_i$.
- (5) Calculate $f_- = f(X^t)$.
- (6) If $f_- < f_c$ set $X^c = X^t$, $f_c = f_-$ and $s_i = -as_i$ proceed from step 1 for the next value of i .
- (7) If $f_- \geq f_c$ calculate an appropriate step by: $s_i = -\frac{1}{2}(f_+ - f_-)/(f_+ + f_- - 2f_c)s_i$.
- (8) Proceed from step 1 for the next value of i .

a , is an enhancement factor chosen by the user. If after looping over all variables there is no progress, a line search is performed in the direction $s = (s_1, s_2, \dots, s_n)$. The above procedure is repeated until a preset number of calls to the objective function is reached, whereupon control is transferred to the calling program. The routine is terminated also when a preset number of failures is reached. We consider as a failure the case where either looping over all variables or after having performed the line search, the relative progress made (i.e. $|f_{\text{initial}} - f_{\text{final}}|/|f_{\text{initial}}|$) is less than a threshold.

References

- [1] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile, September 1994.
- [2] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proceedings of the 11th International Conference on Data Engineering, Taiwan, March 1995, pp. 3–14.
- [3] S. Amo, D.A. Furtado, First-order temporal pattern mining with regular expression constraints, Data and Knowledge Engineering 62 (3) (2007) 401–420.
- [4] J. Ayres, J. Gehrke, T. Yiu, J. Flannick, Sequential pattern mining using bitmaps, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Canada, 2002, pp. 429–435.
- [5] S.D. Bay, M.J. Pazzani, Detecting group differences: mining contrast sets, Data Mining and Knowledge Discovery 5 (3) (2001) 213–246.
- [6] R.J. Bayardo Jr., Brute-force mining of high-confidence classification rules, in: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, 1997, pp. 123–126.
- [7] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The Protein Data Bank, Nucleic Acids Research 28 (2000) 235–242.
- [8] F. Bonchi, C. Lucchese, Extending the state-of-the-art of constraint-based pattern discovery, Data and Knowledge Engineering 60 (2) (2007) 377–399.
- [9] S. Chakrabartty, G. Cauwenberghs, Forward decoding kernel machines: a hybrid HMM/SVM approach to sequence recognition, in: Proceedings of the International Workshop on Pattern Recognition with Support Vector Machines, Lecture Notes in Computer Science, vol. 2388, 2002, pp. 278–292.
- [10] T.-Z. Chen, S.-C. Hsu, Mining frequent tree-like patterns in large databases, Data and Knowledge Engineering 62 (1) (2007) 65–83.
- [11] C. Ding, I. Dubchak, Multi-class protein fold recognition using support vector machines and neural networks, Bioinformatics 17 (2001) 349–358.
- [12] G. Dong, J. Li, Efficient mining of emerging patterns: discovering trends and differences, in: Proceedings of the Knowledge Discovery in Databases (KDD-99), San Diego, CA, USA, 1999, pp. 43–52.
- [13] G.A. Evangelakis, J.P. Rizos, I.E. Lagaris, I.N. Demetropoulos, MERLIN – A portable system for multidimensional optimization, Computer Physics Communications 46 (1987) 401–415.
- [14] T.P. Exarchos, C. Papatoukas, C. Lampros, D.I. Fotiadis, Mining sequential patterns for protein fold recognition, Journal of Biomedical Informatics 44 (1) (2008) 165–179.
- [15] T.P. Exarchos, C. Papatoukas, C. Lampros, D.I. Fotiadis, Protein classification using sequential pattern mining, in: Proceedings of the IEEE Engineering in Medicine and Biology Conference, New York, USA, 2006, pp. 5814–5817.
- [16] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery: an overview, in: Advances in Knowledge discovery and Data Mining, AAAI Press/MIT Press, 1996, pp. 1–36.
- [17] M. Garofalakis, R. Rastogi, K. Shim, SPIRIT: sequential pattern mining with regular expression constraint, in: Proceedings of the 25th International Conference on Very Large Databases, 1999, pp. 223–234.

- [18] J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.
- [19] J. Hu, S.G. Lim, M.K. Brown, Writer independent on-line handwriting recognition using an HMM approach, *Pattern Recognition* 33 (2000) 133–147.
- [20] S. Jaillet, A. Laurent, M. Teisseire, Sequential patterns for text categorization, *Intelligent Data Analysis* 9 (2006) 1–16.
- [21] X. Ji, J. Bailey, G. Dong, Mining minimal distinguishing subsequence patterns with gap constraints, *Knowledge and Information Systems* 11 (3) (2007) 259–286.
- [22] Joachims Text categorization with support vector machines: learning with many relevant features, in: *Proceedings of the ECML-98, 10th European Conference on Machine Learning*, Chemnitz, DE, Springer-Verlag, Heidelberg, DE, 1998, pp. 137–142.
- [23] H.-C. Kum, J.H. Chang, W. Wang, Benchmarking the effectiveness of sequential pattern mining methods, *Data and Knowledge Engineering* 60 (1) (2007) 30–50.
- [24] C. Lampros, C. Papaloukas, T.P. Exarchos, Y. Goletsis, D.I. Fotiadis, Sequence-based protein structure prediction using a reduced state-space hidden Markov model, *Computers in Biology and Medicine* 37 (9) (2007) 1211–1224.
- [25] N. Lesh, M.J. Zaki, M. Ogihara, Mining features for sequence classification, in: *Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, San Diego, CA, 1999, pp. 342–346.
- [26] N. Lesh, M.J. Zaki, M. Ogihara, Scalable feature mining for sequential data, *IEEE Intelligent Systems* 15 (2) (2000) 48–56.
- [27] M. Li, R. Sleep, A robust approach to sequence classification, in: *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*.
- [28] B. Liu, W. Hsu, Y. Ma, Integrating classification and association rule mining, in: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, California, 1998, pp. 80–86.
- [29] D. Loewenstern, H. Berman, H. Hirsh, Maximum a posteriori classification of DNA structure from sequence information, in: *Proceedings of the Pacific Symposium on Biotech*, 1998, pp. 667–668.
- [30] S. Mehta, D. Dinakarpanian, ConsDiff: an algorithm for the detection of conserved differences between protein sequences, *Data and Knowledge Engineering* 53 (2005) 31–43.
- [31] A.G. Murzin, S.E. Brenner, T. Hubbard, C. Chothia, SCOP: a structural classification of proteins database for the investigation of sequences and structures, *Journal of Molecular Biology* 247 (1995) 536–540.
- [32] D.G. Papageorgiou, I.N. Demetropoulos, I.E. Lagaris, MERLIN-3.1.1. A new version of the Merlin optimization environment, *Computer Physics Communications* 159 (2004) 70–71.
- [33] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, Mining sequential patterns by pattern-growth: the prefixSpan approach, *IEEE Transactions on Knowledge and Data Engineering* 16 (2004) 1424–1440.
- [34] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1992.
- [35] L. Rabiner, A tutorial on hidden Markov models and selected application in speech recognition, *Proceedings of the IEEE* 77 (1989) 257–286.
- [36] G. Rätsch, S. Sonnenburg, C. Schäfer, Learning interpretable SVMs for biological sequence classification, *BMC Bioinformatics* 7 (Suppl. 1) (2006) S9.
- [37] R. Srikant, R. Agrawal, Mining sequential patterns: generalizations and performance improvements, in: *Proceedings of the Fifth International Conference on Extending Database Technology, EDBT*, vol. 1057, Springer-Verlag, 1996, pp. 3–17.
- [38] V.S.-M. Tseng, C.-H. Lee, CBS: a new classification method by using sequential patterns, in: *Proceedings of the SIAM International Data Mining Conference*, California, USA, 2005.
- [39] K. Wang, Y. Hu, J. Hu Yu, Scalable sequential pattern mining for biological sequences, in: *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, USA, 2004, pp. 178–187.
- [40] O. Yakhnenko, A. Silvescu, Vasant Honavar, Discriminatively trained Markov model for sequence classification, in: *Proceedings of the Fifth IEEE International Conference on Data Mining*, 2005, pp. 498–505.
- [41] M.J. Zaki, Efficient enumeration of frequent sequences, in: *Seventh International Conference on Information and Knowledge Management*, Washington, DC, 1998, pp. 68–75.
- [42] M.J. Zaki, Sequence mining in categorical domains: incorporating constraints, in: *Proceedings of the Ninth International Conference on Information and Knowledge Management*, USA, 2000, pp. 422–429.



Themis P. Exarchos was born in Ioannina, Greece, in 1980. He received the Diploma Degree in Computer Engineering and Informatics from the University of Patras, in 2003. He is currently working toward the Ph.D. degree in Medical Physics at the University of Ioannina. His research interests include data mining, decision support systems in healthcare, biomedical applications and bioinformatics.



Markos G. Tsipouras was born in Athens, Greece, in 1977. He received the diploma degree and the M.Sc. in computer science from the University of Ioannina, Greece, in 1999 and 2002, respectively. He holds a Ph.D. degree in the Automated Diagnosis of Cardiovascular Diseases, from the Department of Computer Science at the University of Ioannina. His research interests include biomedical engineering, decision support and medical expert systems and biomedical applications.



Costas Papaloukas was born in Ioannina, Greece, in 1974. He received the diploma degree in computer science and the Ph.D. degree in biomedical technology from the University of Ioannina, Ioannina, Greece, in 1997 and 2001, respectively. He is an Assistant Professor of Bioinformatics with the Department of Biological Applications and Technology, University of Ioannina. His research interests include biomedical engineering and bioinformatics.



Dimitrios I. Fotiadis was born in Ioannina, Greece, in 1961. He received the Diploma degree in chemical engineering from National Technical University of Athens, Greece, and the Ph.D. degree in chemical engineering from the University of Minnesota, Twin Cities. Since 1995, he has been with the Department of Computer Science, University of Ioannina, Greece, where he currently is an Associate Professor. He is the director of the Unit of Medical Technology and Intelligent Information Systems. His research interests include biomedical technology, biomechanics, scientific computing, and intelligent information systems.