Knowl Inf Syst
DOI 10.1007/s10115-008-0146-2

REGULAR PAPER

An optimized sequential pattern matching methodology for sequence classification

Themis P. Exarchos · Markos G. Tsipouras · Costas Papaloukas · Dimitrios I. Fotiadis

Received: 25 September 2007 / Revised: 7 March 2008 / Accepted: 12 April 2008

Abstract In this paper we present a novel methodology for sequence classification, based on sequential pattern mining and optimization algorithms. The proposed methodology automatically generates a sequence classification model, based on a two stage process. In the first stage, a sequential pattern mining algorithm is applied to a set of sequences and the sequential patterns are extracted. Then, the score of every pattern with respect to each sequence is calculated using a scoring function and the score of each class under consideration is estimated by summing the specific pattern scores. Each score is updated, multiplied by a weight and the output of the first stage is the classification confusion matrix of the sequences. In the second stage an optimization technique, aims to finding a set of weights which minimize an objective function, defined using the classification confusion matrix. The set of the extracted sequential patterns and the optimal weights of the classes comprise the sequence classification model. Extensive evaluation of the methodology was carried out in the protein classification domain, by varying the number of training and test sequences, the number of patterns and the number of classes. The methodology is compared with other similar sequence classification approaches. The proposed methodology exhibits several advantages, such as automated weight assignment to classes using optimization techniques and knowledge discovery in the domain of application.

Keywords Sequential pattern mining · Sequential pattern matching · Sequence classification · Optimization

T. P. Exarchos
Department of Medical Physics, Medical School, University of Ioannina, 45110 Ioannina, Greece

M. G. Tsipouras · D. I. Fotiadis (✉)
Department of Computer Science, Unit of Medical Technology and Intelligent Information Systems, University of Ioannina, PO Box 1186, 45110 Ioannina, Greece
e-mail: fotiadis@cs.uoi.gr

C. Papaloukas
Department of Biological Applications and Technology, University of Ioannina, 45110 Ioannina, Greece

1 Introduction

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, bio-sequences (DNA, proteins), web-usage data, multiplayer games, and plan-execution traces. Classification is the procedure in which given a collection of training records, each one containing a set of attributes, with one of them being the class, to find a model for the class attribute as a function of the values of other attributes. The result of the classification is that new records are assigned to a class as accurately as possible.

Sequence classification is an important problem that arises in many real-world applications, such as protein function prediction, text classification, speech recognition, intrusion detection, etc. [1]. 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 the sequence. Data mining and machine learning algorithms offer an effective approach to design sequence classifiers, when a training set of labeled sequences is available [2].

The problem of sequence classification has been addressed in the past; however, it has not received too much attention. The earliest approaches employed hidden Markov models [3], finite automata and entropy based approaches [4]. The most recent techniques treat the problem of sequence classification as a feature mining problem [5–7], i.e. they mine sequential patterns from a set of training sequences and then use these patterns for classification. The FeatureMine algorithm uses these patterns as features; in this way the sequences are vectorized based on the matched sequential patterns and then standard classification algorithms such as naïve Bayes and winnow are applied to the vectorized sequences [5,6]. The Classify by Sequences (CBS) algorithm mines sequential patterns from the sequences and then assigns a score to each sequence for each class, using a scoring function [7], which is based on the length of the matched sequential patterns. Tseng et al. [7] presents two approaches, the CBS_ALL and the CBS_CLASS. Experimental results show that CBS_CLASS outperforms both CBS_ALL and FeatureMine [7].

Recently, many data mining techniques like association rules, sequential patterns, clustering and classification, emerged in various research topics [8–12]. Most of the existing data mining methods are designed for solving some specific problem independently. 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 the data mining tasks. For example, the CBA [13] method delivers higher accuracy than traditional classification methods such as C4.5 [11]. Hence, it is a promising direction to integrate different kinds of data mining methods to form a new methodology for solving complex data mining problems.

In this work we propose a novel methodology for sequence classification that is able to work in many sequential domains. 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 from 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 classes. The proposed methodology

extends a previously reported sequence classification algorithm [7] by introducing a set of weights and obtaining optimal values for them using optimization algorithms.

The proposed methodology 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 sets of sequential patterns, and their tuning through optimization, during the second stage, which is proposed for the first time in the literature. The methodology, through the optimization stage, assigns optimal values for these weights to improve the sequence classification performance. The introduction and optimization of the weights is motivated by the fact that the sequential patterns, extracted from the sequences, do not describe all classes with the same adequacy; some classes are over described from the sequential patterns while for some others this description is rather poor. The weights are introduced to balance this trend, and subsequently, an optimization technique is used to automatically calculate optimal values for them. In addition, the weights integrate the available information for each class, since the description of classes with more information (i.e. having a large number of training sequences) is more reliable. The results indicate that the employment of the optimal weights highly increases the classification accuracy of the simple sequential pattern based classifier. Furthermore, the methodology provides to the domain experts knowledge for their domain (by means of the extracted sequential patterns and the optimal weights). Finally, the proposed methodology is generic and can incorporate different algorithms/approaches in any of its stages.

The rest of the manuscript is organized as follows. In Sect. 2, the two stages of the proposed methodology are described in detail. Section 3 presents the experiments from the protein classification domain used to evaluate the methodology, the comparative work and the obtained results, for all experiments. Finally, in the discussion section, qualitative conclusions are derived and quantitative comments concerning the obtained results are addressed.

2 Methods

The proposed methodology includes two stages (Fig. 1). In the first stage, a sequence classification methodology is defined. For the realization of the first stage, a dataset $D = \{S_i, c_i\},\ i = 1, \ldots, l_S$, where S_i is a sequence and c_i is its class, with l_c different classes ($c_i = \{1, \ldots, l_c\}$) and l_S is the number of sequences in the dataset ($|D| = l_S$), and a vector of class weights $w(|w| = l_c)$ are required. This stage is realized in four steps and its outcome is the classification confusion matrix. The second stage is an optimization technique which is based on an objective function, defined by the classification confusion matrix, aiming to find a set of weights, w, which minimize the objective function.

2.1 Stage 1: Sequence classification

Figure 2 presents the pseudo code for the realization of the four steps of the first stage. The list of symbols used and their explanation is presented in Table 1.

Step 1 Sequential pattern mining

The training sequences are divided into l_c subsets, each one containing all sequences belonging to the same class $\left(S^j, j = 1, \ldots, l_c\right)$. Then, sequential pattern mining (*SPM*) [9] is applied to each subset, generating l_c sets of sequential patterns $\left(P^j\right)$, satisfying the

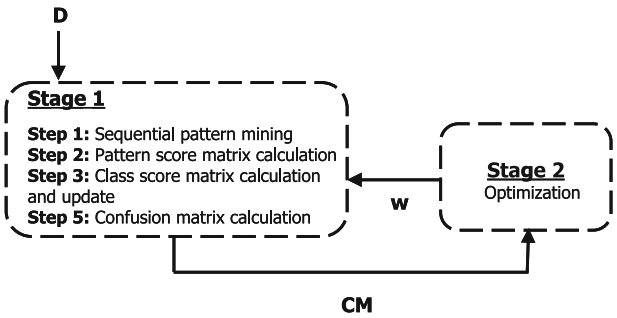

Fig. 1 The two-stage methodology

Table 1 List of symbols used and their explanation

Symbol	Explanation
$D = \{S_i, c_i\}$	Database of sequences
l_S	Overall number of sequences in D, ($\|D\|$)
S_i	The ith sequence
c_i	Class of the ith sequence
l_c	Number of classes
w	Vector of class weights. $\|w\| = l_c$
S^j	The subset of sequences belonging to the jth class
SPM	Sequential pattern mining procedure
P^j	The jth set of sequential patterns. extracted from S^j
PSM^j	The jth pattern score matrix for the P^j
P_m^j	The mth pattern of the jth set of patterns
CSM	The class score matrix
pc_i	The predicted class of the ith sequence
CM	The confusion matrix
w^*	The optimal vector of class weights
D_{train}	The training set
D_{test}	The test set
I	Different items in the sequences (alphabet)
min_sup	Minimum Support (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

user-defined constraints. The above is followed by the CBS_CLASS [7] algorithm reported to outperform CBS_ALL, who mines the whole database of sequences for sequential patterns. This step closely resembles the feature mining problem [5,6]. 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, $j = 1, \ldots, l_c$, characterizing the l_c classes.

In the *SPM* procedure, we can incorporate several constraints, that allow for flexible gap of the extracted sequential patterns. Several algorithms have been reported in the literature which implement the *SPM* procedure [9,14–16]. However, little work has been done on constrained *SPM* [17–20]. An algorithm that performs efficient and effective constrained *SPM* is the cSPADE algorithm [17]. The cSPADE algorithm is based on the SPADE algorithm [16] which 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 [18,19].

Step 2 Pattern score matrix calculation

After the extraction of the sequential patterns, for each class we create a pattern score matrix PSM^j, $j = 1, \ldots, l_c$ (a PSM matrix is created for each class). 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: if the P_m^j pattern i.e. the mth pattern of the jth class is contained in the S_i sequence then the (m, i) element of the PSM^j matrix is equal to the value length$(P_m^j) - 1$ divided by the number of patterns describing the jth class. If the P_m^j pattern is not contained in the S_i sequence then $PSM^j\ (m, i)$ is equal to 0.

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 describing a class, the more significant is each one of these patterns. The above scoring function assigns higher scores to the longer sequential patterns, by adding their score along with the score of all their subsequences. This has been taken into consideration since, longer sequential patterns are much more important than shorter ones.

Step 3 Class score matrix calculation and update

From the PSM^j matrices, we derive the class score matrix (CSM). Each (j, i) element of this matrix is the score of the ith sequence for the jth class, and it is defined as the sum of the scores of all patterns belonging to the jth class for the ith sequence. This is shown schematically in Fig. 3. 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; thus, the size of the CSM matrix is $l_c \times l_S$. Then, 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: $\text{row}_j = w\,(j) \cdot \text{row}_j$, where row_j is the jth row of the CSM matrix and $w\,(j)$ is the jth element of the class weight vector.

Step 4 Confusion matrix calculation

For each sequence S_i, a class is predicted (pc_i), based on the updated CSM matrix. This predicted class is defined as the class that obtains the highest score for the ith sequence: $pc_i = \arg\max_{j=1,\ldots,l_c} (CSM\,(j, i))$. Based on the real class c_i and the predicted class pc_i, the confusion matrix (CM) for all the sequences is calculated, as it is shown in step 4 of the pseudocode (Fig. 2).

STEP 1: Sequential pattern mining (*SPM*)

- *for* $j = 1,...,l_c$ (for each class)
  - $S^j = \{S_i | c_i = j\}$

    (select all sequences from the dataset that belong to the j^{th} class)
  - $P^j = SPM\left(S^j\right)$

    (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: Pattern Score Matrix (PSM) calculation

- *for* $j = 1,...,l_c$ (for each class)
  - *for* $m = 1,...,l_j$ (for each pattern of this class, $\left|P^j\right| = l_j$)
    - *for* $i = 1,...,l_S$ (for each sequence in the dataset)
      - *if* P_m^j is contained in S_i *then* $PSM^j(m,i) = \left(length\left(P_m^j\right)-1\right) \Big/ \left|P^j\right|$ *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 it is assigned the value of $length\left(P_m^j\right)-1$ divided by the number of patterns describing the j^{th} class, else it is assigned the value of 0).
      - *end*
    - *end*
  - *end*
- *end*

STEP 3: Class Score Matrix (CSM) calculation and update

- *for* $j = 1,...,l_c$ (for each class)
  - *for* $i = 1,...,l_S$ (for each sequence in the dataset)
    - $CSM(j,i) = \sum_{m=1}^{l_j} PSM^j(m,i)$
    - $CSM(j,i) = wc(j) \cdot CSM(j,i)$

      (The CSM value for the i^{th} sequence for the j^{th} class, which denotes the score of 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. Each element of the columns of the *CSM* is updated, by multiplying it with a class weight).
  - *end*
- *end*

STEP 4: Confusion matrix (*CM*) calculation

- *for* $i = 1,...,l_S$ (for each sequence in the dataset)
  - $pc_i = \arg\max_{j=1...l_c}\left(CSM(j,i)\right)$

    (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)
  - $CM\left(c_i, pc_i\right) = CM\left(c_i, pc_i\right) + 1$
- *end*

Fig. 2 Pseudocode which describes the first stage of the proposed methodology

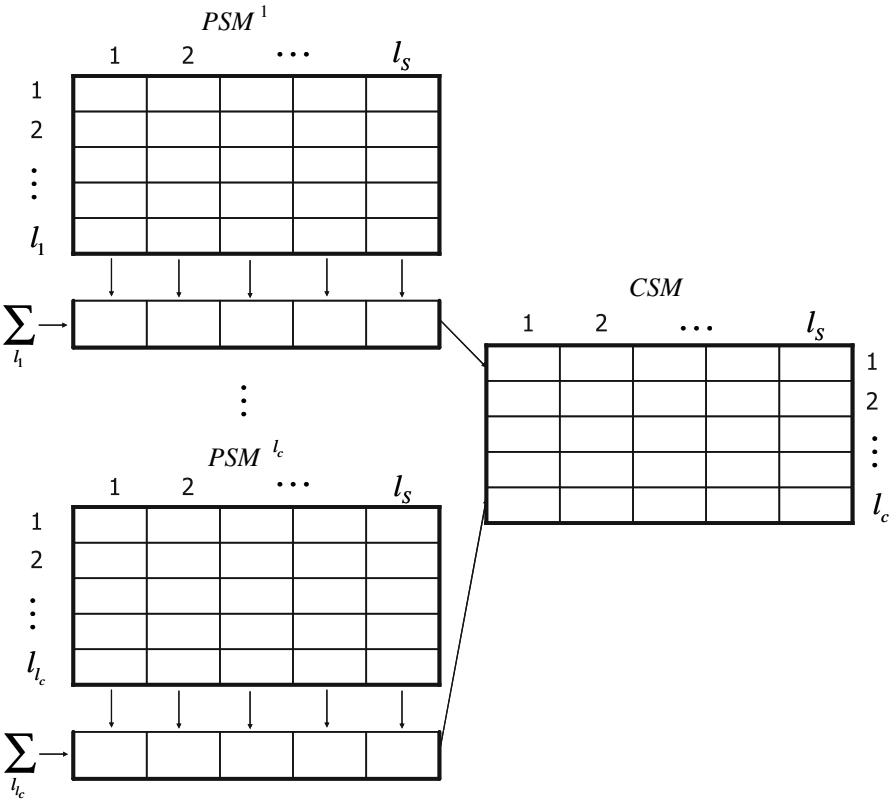

Fig. 3 Calculation of the class score matrix (*CSM*) from the pattern score matrix (*PSM*). The columns of each *PSM* are summed in order to create the rows of the *CSM*

2.2 Stage 2: Optimization

In the second stage, we try to calculate a set of weights w that derive the highest classification accuracy of the sequences in the dataset. Initially, a cost function is defined, based on the CM:

$$f\,(D, w) = l_S - \text{trace}\,(CM)\,. \tag{1}$$

Equation (1) can be formulated as an optimization problem, minimizing $f\,(D, w)$ with respect to w. Different values for the weights have an impact on the cost function, since they affect CM. This cost function has been selected since its minimum value is 0, and this value is obtained if (with the appropriate w) all sequences are correctly classified ($l_S = \text{trace}\,(CM)$). Local optimization strategy is preferred, since an initial point is available ($w = 1$) and it is significantly faster than global optimization. In addition, it is very difficult to calculate analytically the derivatives of the objective function. Based on the above, we employed the Nelder-Mead simplex search method [21] to solve the optimization problem. This is a direct search method for multidimensional unconstrained minimization which does not use numerical or analytic gradients.

The Nelder-Mead simplex search method initially defines a simplex $\Theta \in R^n$ in the—n dimensional space, which is characterized by the $n+1$ distinct vectors which are its vertices.

At each step of the search, a new point in or near the current simplex is generated. The function value at the new point is compared with the function's values at the vertices of the simplex and, usually, one of the vertices is replaced by the new point, giving a new simplex. This step is repeated until the diameter of the simplex is less than a specified tolerance. The result of the optimization procedure is a set of optimal class weights w^*.

3 Results

The proposed methodology has been evaluated using an appropriate sequence dataset. Results of the proposed methodology are presented for both cases with and without the use of the optimization stage ($\mathbf{w} = 1$) [22,23]. Our methodology is also compared with a previously reported sequence classification method, the CBS algorithm [7].

3.1 Dataset

The sequence classification domain that was selected is the classification of protein primary structures into folds and classes. The formulation of *SPM*covers almost any categorical sequential domain [24]. 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 were taken from the Protein Data Bank (PDB) [25]. All members of this group correspond to a specific fold of the Structural Classification of Proteins (SCOP) database [26]. Specifically the 17 SCOP folds, with at least 30 members, from classes A and B were used to derive the training and test data. From the resulted 1,000 proteins, the two thirds from each category were used for training, while the rest for evaluation.

3.2 Evaluation

From the above sequences, four datasets were derived and four different classification experiments were performed. Each dataset was divided into training and test sets.

- In the first experiment (Exp. 1) we used 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 7 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 2 categories (All sequences of class A folds belong to the first category and all sequences of class B folds belong to the second category). A total of 666 proteins form the training set and 334 proteins form the test set.

In all experiments, we set the minimum support to 50%, meaning that a pattern should be present in at least half of the training sequences, as values greater than that should

indicate a propensity towards the correct description of this class. Also, for each of the above experiments, we varied the number of max_gap from 1 to 5, since values greater than 5 made the number of the extracted sequential patterns prohibitively large. Thus we created 20 experiments; in all experiments, the training set was used for sequential pattern mining and for calculating w^*. In the testing phase, the sequential patterns and w^* are used to classify the sequences of the test set.

3.3 Comparative work

The CBS algorithm has also been tested using the same experimental procedure. We have implemented the CBS_CLASS variation of the CBS algorithm. In the CBS_CLASS

Table 2 Number of extracted sequential patterns and accuracy results for the training and test sets for the three approaches (CBS, SPM, OSPM)

max_gap	# Patterns		CBS[a]	SPM[b]	OSPM[c]		CBS	SPM	OSPM
Exp. 1: $\|D_{\text{train}}\| = 666$, $\|D_{\text{test}}\| = 334$ and $l_c = 17$									
1	1568	Training	36.5	36.5	40.8	Test	22.2	22.5	22.5
2	3670		32.0	38.4	60.8		19.2	18.3	32.0
3	7404		22.7	54.4	65.8		14.1	27.0	38.0
4	17542		33.8	69.1	77.9		22.5	35.9	41.3
5	38557		42.5	67.6	78.1		24.6	30.5	39.2
Exp. 2: $\|D_{\text{train}}\| = 406$, $\|D_{\text{test}}\| = 203$ and $l_c = 10$									
1	1142	Training	43.8	44.1	66.8	Test	28.1	30.1	38.9
2	2444		28.1	34.2	65.8		16.8	18.2	38.4
3	5035		15.3	61.6	70.9		12.8	33.5	41.9
4	12456		22.9	78.6	82.8		19.2	43.8	46.3
5	27603		31.8	78.3	83.0		23.7	40.9	42.4
Exp. 3: $\|D_{\text{train}}\| = 260$, $\|D_{\text{test}}\| = 131$ and $l_c = 7$									
1	426	Training	59.2	59.2	63.9	Test	42.8	42.8	46.6
2	1226		61.2	67.3	78.1		37.4	33.6	43.5
3	2369		65.4	61.9	77.3		50.4	42.0	51.9
4	5086		71.9	75.0	85.4		53.4	44.3	53.4
5	10954		73.5	74.6	84.6		51.2	42.0	53.4
Exp. 4: $\|D_{\text{train}}\| = 666$, $\|D_{\text{test}}\| = 334$ and $l_c = 2$									
1	1568	Training	61.4	62.3	82.1	Test	59.6	60.2	76.1
2	3670		51.2	53.0	85.3		49.4	51.2	75.8
3	7404		49.1	66.8	84.4		48.5	62.0	77.5
4	17542		56.8	71.6	84.5		55.7	65.9	78.4
5	38557		62.2	71.0	84.8		58.4	65.6	79.0

$|D_{\text{train}}|$ and $|D_{\text{test}}|$ denote the sizes of the corresponding training and test sets, respectively
[a]CBS: Accuracy of the classify by sequence algorithm
[b]SPM: Accuracy of the approach without the use of the optimization stage. Sequential pattern matching
[c]OSPM: Accuracy of the proposed methodology. Optimized sequential pattern matching

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 for each sequence is calculated, 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 the value length$(P_m^j)\big/\sum_{m=1}^{l_j}$ length(P_m^j), else it is equal to 0. After the creation of the PSM matrix, the CBS_CLASS algorithm employs the same steps with the stage 1 of the proposed methodology, for the creation of the CSM matrix and the classification of a sequence in a predicted class.

In addition, we compared our method with a well known and widely used method for sequence classification, which is based on hidden Markov models, the Sequence Alignment and Modeling method, SAM [27,28]. SAM employs the Baum-Welch algorithm [29] for training a hidden Markov model and classifies sequences using two approaches: either ranking of the scores obtained for each sequence (SAM_1) or ranking of the E-values obtained for each sequence (SAM_2). Currently, we tested both SAM_1 and SAM_2, with the same training and test sets with the proposed methodology.

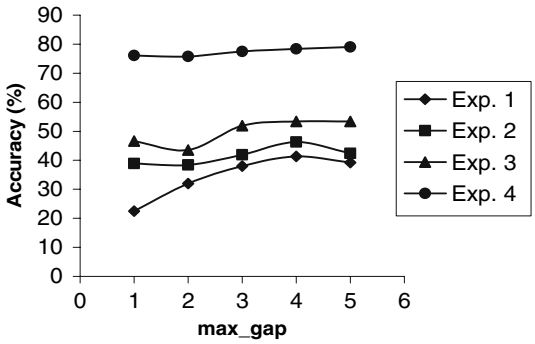

Fig. 4 Graphical representation of the accuracy of the proposed methodology for the four experiments for various values of *max_gap*. The best accuracy for Exp. 1, Exp. 2 and Exp. 3 is obtained for *max_gap*=4 while for Exp. 4, the best accuracy is obtained for *max_gap*=5

Table 3 Accuracy results (%) for SAM_1, SAM_2, CBS, SPM and OSPM in test sequences

Exp. 1: $\lvert D_{\text{train}}\rvert = 666$, $\lvert D_{\text{test}}\rvert = 334$ and $l_c = 17$				
SAM_1	SAM_2	CBS	SPM	OSPM
29.4	35.0	24.6	35.9	41.3
Exp. 2: $\lvert D_{\text{train}}\rvert = 406$, $\lvert D_{\text{test}}\rvert = 203$ and $l_c = 10$				
36.5	40.4	23.7	43.8	46.3
Exp. 3: $\lvert D_{\text{train}}\rvert = 260$, $\lvert D_{\text{test}}\rvert = 131$ and $l_c = 7$				
42.0	42.8	53.4	44.3	53.4
Exp. 4: $\lvert D_{\text{train}}\rvert = 666$, $\lvert D_{\text{test}}\rvert = 334$ and $l_c = 2$				
59.0	63.8	58.4	65.6	79.0

The accuracies of CBS, SPM and OSPM are derived as the maximum ones for each different experiment

3.4 Classification performance

Table 2 presents the obtained results for all experiments and for all the different values of the *max_gap*. It is mentioned that the accuracy results are presented both for the training and the test sets. In the first experiment, the number of patterns varies from 1,568 to 38,557 for *max_gap* = 1–5. Similarly, in the second experiment the number of patterns varies from 426 to 10,954, in the third experiment this number varies from 1,142 to 27,603 while for the fourth experiment, the number of patterns varies as in the first experiment since the patterns from the 17 classes were used as patterns for the two classes. Figure 4 presents the classification accuracies (for the test set) for all 4 experiments and for all 5 different values of *max_gap*. Table 3 presents the accuracy results in the test sequences for SAM_1, SAM_2, CBS, SPM and OSPM. It should be mentioned that in Table 3, the accuracies of CBS, SPM and OSPM are the maximum ones for each different experiment (Exp. 1, Exp. 2, Exp. 3 and Exp. 4) for the corresponding *max_gap* value of Table 2.

4 Discussion

In the current work we presented a novel methodology for the automated generation of sequence classification models. 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 class are calculated, using an optimization technique. The obtained optimal class weights along with the extracted sequential patterns compose the sequence classification model, which is used to classify the test sequences.

The proposed methodology introduces several innovative features. To our knowledge, the automatic assignment of weights to sets of sequential patterns 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 [5,6] to standard classification algorithms, or employ a scoring function, similar to the one reported in the current work [7]. The weight assignment 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, superior to previous works. Also, the results of the simple sequential pattern based classifier are significantly improved, when the optimal weights are applied.

The methodology can be applied to other domains of application; the application of the methodology is straightforward, since only fundamental information related to the domain of application is needed: the dataset, consisting from the set of sequences, the class of each sequence, and the alphabet that is used to form the sequences. For example, in the domain of document classification where there exists a set of documents, each of them belonging to a predefined class, our methodology can be applied as follows: the alphabet is defined as the total number of (stemmed) words existing in all documents and thus, each document is considered as a sequence of words. Then, based on the class of each document, the set of documents is divided into subsets, where each subset contains documents of the same class. After that, sequential pattern mining (step 1 of the first stage) is applied to each of the subsets of documents (sequences) to generate a set of sequential patterns describing each one of the document classes. Steps 2–4 of the first stage are then applied. In these steps, each set of sequential patterns describing a document class is considered of equal importance with all others; the importance of each set is defined through the introduction of the weights (one weight for each class) and initially the values of all weights are set to 1 (i.e. they are

considered of equal importance). Finally, Stage 2 is applied to assign optimal values to each weight in order to increase the correctly classified documents. This is performed by defining an appropriate objective (error) function, in our case based on the document classification confusion matrix, and then minimizing its value using an optimization technique/strategy.

Additionally, the methodology is generic since different components can be used for any of its stages, i.e. different SPM algorithm, alternative objective function and/or optimization method (local or global). Also, the scoring function could be modified to integrate different preferences, e.g. in case the sequences are composed by itemsets with multiple items, the scoring function could be modified to use either the *length* (number of itemsets) or the *size* (number of items) of the pattern in the numerator (currently, itemsets are composed by single items and thus the *length* and the *size* of a pattern are the same) of the scoring function.

The *SPM* approach, employed in this work, is suitable for analyzing sequences and 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 as 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. Its basic design (i.e. provide higher score to sequential patterns of higher length by adding their score along with the score of all their subsequences) was obtained from the CBS algorithm. In addition, we utilized also as scoring function (in the numerator) the times a sequential pattern is contained in the sequence raised in the power of $n (n = 1, 2, \ldots)$, the logarithm of the length of the pattern, the length of the pattern raised in the power of $n (n = 1, 2, \ldots)$, the support of the pattern and others. All the above, including the scoring function of the CBS algorithm, reported lower classification results when they were used in our sequence database. More specifically (Table 2), the accuracy obtained in almost all classification problems (in 19 out of 20 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 15 out of 20 experiments). The average accuracy for all experiments is 46.1% for training using the CBS algorithm and 61.3% using SPM (improvement of 15.2%), while the average accuracy in testing is 35.5% for the CBS and 40.5% for SPM (improvement of 5%).

The proposed methodology has been evaluated systematically, using 20 different evaluation experiments (four datasets multiplied by five different values of the *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 38,557. This large number of different evaluation experiments resulting from the wide range of parameters, ensures the reliable evaluation of the proposed methodology in protein classification domain.

Comparing the computational complexity and the running times of the three algorithms (SPM, OSPM and CBS), SPM and CBS have the same computational complexity and running time both in training and in testing, since for every different value of max_gap, the same number of sequential patterns are both mined in training and matched in testing. OSPM presents higher computational complexity and running time than the other two algorithms in the training, since it employs two stages, the first stage which is common with SPM and

CBS and the second stage, which employs an optimization technique. Thus, the additional complexity and running time of the OSPM is due to the optimization stage and depends on the selection of the optimization technique. In the current work, the Nedler-Mead simplex search method is employed, which is a local optimization technique with low computational complexity. This complexity and training time depend on the number of classes, since its value defines the number of parameters for optimization. The running time for testing depends only on the classification difficulty of the experiment (number of patterns, number of classes, number of test sequences) and thus all three algorithms report the same running time in terms of each different experiment. It should be mentioned that the number of extracted sequential patterns, highly depends on the max_gap value, thus as max_gap increases the number of sequential patterns and subsequently the running time for all three algorithms, increase.

In addition, the proposed optimization stage 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 20 experiments, both in training and testing, the proposed methodology (mentioned as OSPM in Table 2) presents higher accuracy. The average accuracy for all experiments is 75.2% in training and.50.8% in testing, improving the accuracy of the CBS in training by 29.1% and SPM (which is the first stage of the OSPM methodology) by 13.9%. The respective improvement in testing is 15.3% and 10.3%, compared to the CBS and the SPM, respectively. It should be noted that the best accuracy for OSPM in Exp. 1,2,3 experiments is achieved for $max\_gap=4$, while for Exp. 4, the best accuracy is achieved for $max\_gap=5$ (Fig. 4). Thus, patterns with up to 3 intervening amino-acids between two consecutive amino-acids constituting a pattern are the most descriptive in the cases of Exp. 1, Exp. 2 and Exp. 3, while patterns with up to 4 intervening amino-acids between two consecutive amino-acids, are more suitable for Exp. 4. This can be attributed to the higher homology between the folds in Exps. 1–3, since Exps. 1–3 are fold prediction problems. On the other hand, Exp. 4 is a class prediction problem, and between classes, there exists lower homology. Thus, a lower value of max_gap (4) is more appropriate for (higher homology) fold recognition, while for (lower homology) class prediction, a higher value of max_gap (5) reports the best results.

The proposed methodology is compared with the Sequence Alignment and Modeling system [27,28], which is a well known and widely used method for sequence classification that is based on hidden Markov models (Table 3). The comparison shows that the proposed methodology outperforms both SAM_1 and SAM_2 in terms of accuracy in the test sequences, for all experiments. The average accuracy in Table 3 is 41.7, 45.5, 40, 47.4 and 55% for SAM1, SAM2, CBS, SPM and OSPM, respectively. It should be mentioned that the methodology and all comparative methods, provide low classification results in terms of "raw accuracy numbers" in Exp. 1, Exp. 2 and Exp. 3. However, the performance of all methods becomes evident when compared with the respective performance of a classifier that makes random predictions. In this case, the proposed methodology and all comparative methods report an accuracy of more than 50% accuracy in Exp.4, where there are two classes, and the accuracy of a classifier that makes random predictions is 50%. In Exp.1, Exp. 2 and Exp. 3, the number of classes is 17, 10 and 7 respectively, and thus the accuracy of a classifier that makes random predictions is 5.89, 10 and 14.29%, respectively. It is worth mentioning that, the relative literature presents comparable results when the number of the target classes (folds) for prediction is similar [30].

The proposed methodology is based on sequential pattern mining. A drawback of the methodology is that a large number of patterns is discovered which increases exponentially with *max_gap*. Although, the extraction of sequential patterns is relatively fast (due to the cSPADE algorithm), the overall processing time increases. 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; this feature will be addressed in the future. Additionally, the optimization stage, although significantly improves the classification accuracy of the approach without the optimization stage, increases the computational effort and the overall time for the training. For this reason, a local optimization strategy was selected, which however does not ensure the best results.

5 Conclusions

A novel sequence classification methodology has been presented along with an extensive evaluation in the domain of protein classification and the obtained results indicate its effectiveness. Application of the methodology in other discrete sequential domains will fully reveal its potential. Our work in the future must focus on: (1) using different techniques for sequential pattern mining (i.e. mine sequential patterns with sequence alignment [31,32]), (2) using methods for sequential pattern selection, or the use of specific types of patterns like closed [33] or maximal [34] sequential patterns, or minimal distinguishing sequential patterns [35], (3) employing different scoring function, and/or optimization strategies and (4) extending the methodology in order to handle time series.

References

1. Fayyad UM, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery: an overview. In: Advances in knowledge discovery and data mining. AAAI Press/MIT Press, Cambridge, pp 1–36
2. Han J, Kamber M (2000) Data mining: concepts and techniques. Morgan Kaufmann, Menlo Park
3. Rabiner L (1989) A tutorial on hidden Markov models and selected application in speech recognition. Proc IEEE 77:257–286
4. Loewenstern D, Berman H, Hirsh H (1998) Maximum a posteriori classification of DNA structure from sequence information. In: Proceedings of Pacific symposium in Biotech, pp 667–668
5. Lesh N, Zaki MJ, Ogihara M (1999) Mining features for sequence classification. In: 5th ACM SIGKDD international conference on knowledge discovery and data mining (KDD). San Diego, pp 342–346
6. Lesh N, Zaki MJ, Ogihara M (2000) Scalable feature mining for sequential data. IEEE Intell Syst 15(2):48–56
7. Shin-Mu Tseng V, Lee C-H (2005) CBS: a new classification method by using sequential patterns. In: Proceedings of SIAM international data mining conference. California, USA
8. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th international conference on very large databases. Santiago, Chile
9. Agrawal R, Srikant R (1995) Mining sequential patterns. In: Proceedings of the 11th international conference on data engineering, Taiwan, pp 3–14
10. Bayardo Jr RJ (1997) Brute-force mining of high-confidence classification rules. In: Proceedings of the third international conference on knowledge discovery and data mining, pp 123–126
11. Quinlan JR (1992) C4.5: Programs for machine learning, Morgan Kaufman, Menlo Park
12. Zaki MJ (1998) Efficient enumeration of frequent sequences. In: 7th International conference on information and knowledge management, Washington DC, pp 68–75
13. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: Proceedings of fourth international conference on knowledge discovery and data mining. AAAI Press, Menlo Park, pp 80–86 (1998)
14. Pei J, Han J, Mortazavi-Asl B, Wang J, Pinto H, Chen Q, Dayal U (2004) Mining sequential patterns by pattern-growth: the prefixspan approach. IEEE Trans Knowl Data Eng 16:1424–1440
15. Ayres J, Gehrke J, Yiu T, Flannick J (2002) Sequential pattern mining using bitmaps. In: Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining, Canada, pp 429–435

16. Chen T-Z, Hsu S-C (2007) Mining frequent tree-like patterns in large databases. Data Knowl Eng 62(1):65–83
17. Zaki MJ (2000) Sequence mining in categorical domains: incorporating constraints. In: Proceedings of the 9th international conference on information and knowledge management, USA, pp 422–429
18. Srikant, R., Agrawal, R.: Mining sequential patterns: generalizations and performance improvements. In: Proceedings 5th International Conference Extending Database Technology, EDBT. Springer, Heidelberg, vol 1057, pp 3–17 (1996)
19. Garofalakis M, Rastogi R, Shim K (1999) SPIRIT: sequential pattern mining with regular expression constraint. In: Proceedings of the 25th international conference on very large databases, pp 223–234
20. Lin M-Y, Lee S-Y (2005) Efficient mining of sequential patterns with time constraints by delimited pattern growth. Knowl Inf Syst 7:499–514
21. Lagarias JC, Reeds JA, Wright MH, Wright PE (1998) Convergence properties of the Nelder-Mead simplex method in low dimensions. SIAM J. Optim. 9(1):112–147
22. Exarchos TP, Papaloukas C, Lampros C, Fotiadis DI (2007) Mining sequential patterns for protein fold recognition. J. Biomed. Inf. (in press)
23. Exarchos TP, Papaloukas C, Lampros C, Fotiadis DI (2006) Protein classification using sequential pattern mining. In: Proceedings of IEEE engineering in medicine and biology conference, New York, USA, pp 5814–5817
24. Wang K, Hu Y, Hu Yu J (2004) Scalable sequential pattern mining for biological sequences. In: Proceedings of the 13th ACM conference on information and knowledge management, USA, pp 178–187
25. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. Nucleic Acids Res 28:235–242
26. Murzin AG, Brenner SE, Hubbard T, Chothia C (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. 247:536–540
27. Hughey R, Krogh A (1996) Hidden Markov models for sequence analysis: extension and analysis of the basic method. CABIOS 12(2):95–107
28. Karplus K, Karchin R, Shackelford G, Hughey R (2005) Calibrating evalues for hidden Markov models using reverse-sequence null models. Bioinformatics 21:4107–4115
29. Baum LE (1972) An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities 3:1–8
30. Moult J, Fidelis K, Rost B, Hubbard T, Tramontano A (2005) Critical assessment of methods of protein structure prediction (CASP)-round 6. Proteins 61(Suppl 7):3–7
31. Kum, H.-C., Chang, JH., Wang, W.: Intelligent sequential mining via alignment: optimization techniques for very large DB. In: Advances in knowledge discovery and data mining. Lecture Notes in Computer Science, vol 4426, pp 587–597 (2007)
32. Rigoutsos I, Floratos A (1998) Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. Bioinformatics 14(1):55–67
33. Tzvetkov P, Yan X, Han J (2005) TSP: mining top-k closed sequential patterns. Knowl Inf Syst 7:438–457
34. Luo C, Chung SM (2007) A scalable algorithm for mining maximal frequent sequences using a sample. Knowl Inf Syst (in press)
35. Ji X, Bailey J, Dong G (2007) Mining minimal distinguishing subsequence patterns with gap constraints. Knowl Inf Syst 11(3):259–286

Author Biographies

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 Dept. 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.