Multidimensional Sequence Classification based on Fuzzy Distances and Discriminant Analysis

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas

Aristotle University of Thessaloniki, Department of Informatics, Thessaloniki, Greece

email: {tefas,pitas}@aiia.csd.auth.gr

Abstract

In this paper, we present a novel method aiming at multidimensional sequence classification. We propose a novel sequence representation, based on its fuzzy distances from optimal representative signal instances, called statemes. We also propose a novel modified Clustering Discriminant Analysis algorithm minimizing the adopted criterion with respect to both the data projection matrix and the class representation, leading to the optimal discriminant sequence class representation in a low-dimensional space, respectively. Based on this representation, simple classification algorithms, such as the nearest subclass centroid, provide high classification accuracy. A three step iterative optimization procedure for choosing statemes, optimal discriminant sub-space and optimal sequence class representation in the final decision space is proposed. The classification procedure is fast and accurate. The proposed method has been tested on a wide variety of multidimensional sequence classification problems, including handwritten character recognition, time series classification and human activity recognition, providing very satisfactory classification results.

I. Introduction

Sequence classification is used in a wide range of applications, where real-world data can be interpreted as time varying sequences. For example, in health informatics, ECGs are multichannel sequences that can be classified, in order to diagnose heart deceases [1], [2]. In image analysis, an object contour can be

Fig. 1. *a) Human walk described as a sequence of human body pose images, b) handwritten characters 'I' and 'J'.*

considered as a sequence of 2D pixels that can provide shape descriptors for object classification [3]. In a similar way, handwritten optical character recognition can be performed by considering the characters as 2D pixel sequences denoting pen tip trajectory [4]. In human-centered video analysis, human actions are interpreted as sequences of human body poses [5], which are high-dimensional body image masks.

Generally speaking, a multidimensional data sequence $\mathbf{s}$ refers to an ordered list of multidimensional *instances* (sequence samples) $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{N_s}]^T$, $\mathbf{s}_n \in \mathbb{R}^N$. Sequences can be labeled by class labels. For example, a sequence representing a Latin character belongs to one of the 26 letters (classes) $\{A, B, ..., Z\}$, appearing in the Latin alphabet. Given a sequence class label set $\mathcal{L} = \{L_1, ..., L_M\}$ and a sequence $\mathbf{s}$, sequence classification is defined as the task of mapping $\mathbf{s}$ to one of the M classes in $\mathcal{L}$. The main challenges in sequence classification addressed in this paper are described subsequently and are illustrated with examples coming from handwritten character recognition and human action recognition.

- The dimensionality of the instance space $\mathbb{R}^N$ may be very high. Moreover, instances may contain redundant information. Thus, if someone wants to use all available information, the classification task will be computationally expensive and memory consuming. However, several applications require fast classification using a low amount of memory. For example, human body poses in Figure 1a are represented by high-dimensional images. However, if one determines a codebook of all important body poses, each instance could be represented by an integer, denoting the most similar pose in the codebook, obtained by vector quantization [6].
- Inter- and intra-class variations. Sequences belonging to the same class may be quite different from each other, while sequences belonging to different classes may be quite similar, as illustrated by the three handwritten characters in Figure 1b.
- Varying sequence length. Sequences belonging to different classes may differ in length. This may

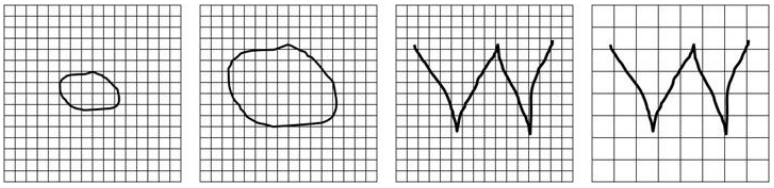

Fig. 2. *Handwritten characters represented as intersection points with a grid.*

Fig. 3. *Walking sequences starting from different human body poses.*

be observed even in sequences belonging to the same class, as illustrated in Figures 2a, 2b, where a handwritten character is represented by a sequence of 2D pixels coordinates $\mathbf{s}_n = [x_n, y_n]^T$, $n = 1, ..., N_s$.

- Variations in speed and sampling. Instances may either be collected using different sampling rates, as shown in Figures 2c, 2d, or sequences describing the same event may be observed having different evolution speed. For example, a person may walk faster than another one.
- Time lag. Periodic sequences belonging to the same class may begin from different starting instances, as illustrated in Figure 3.

Several methods have been proposed in the literature for sequence classification and can be categorized in three categories: distance-based, model-based and feature-based ones [7], [8], [9]. In the following, we provide a comprehensive review of the most important sequence classification methods.

Distance-based methods define a distance function, e.g. the Euclidean distance, to measure the similarity between two sequences. Typically, such distance functions assume that the sequences have the same length and, therefore, are sensitive to sequence length N_s variations. In order to overcome this limitation, sequence alignment techniques have been proposed. Dynamic Time Warping (DTW) [10], [11] aligns two sequences before matching by stretching or shrinking them in length, in order to provide the best match. The Longest Common Subsequence (LCSS) model [12] allows two sequences to stretch, without rearranging the order of their instances and by allowing unmatched instances. These techniques have

high computational complexity and are, usually, computed by dynamic programming. Given a distance function, sequence classification is usually performed by using the K-nearest neighbor (KNN) classifier, as it has been shown that the 1NN classifier is surprising competitive in terms of classification accuracy, compared to other, more complex, classifiers [9]. Traditional KNN classification involves searching the entire training set for the K most similar training sequences to the test sequence and, thus, requires a high computational cost. In order to speed up the search procedure, two approaches have been proposed: building fast search algorithms and/or reducing the training sequence set cardinality. The first approach is implemented by organizing the training set in structures that allow efficient search, e.g. in hierarchical trees [13]. In the latter one, the training set is reduced, by keeping only representative training sequences, which are, subsequently, used by the KNN search algorithm [14].

Model-based classification methods assume that sequences belonging to a class are generated by an underlying probabilistic model. Each class is represented by such a model, which is learned using the training sequences. A test sequence is assigned to the class model that provides the highest likelihood. Naive Bayes classifiers [15] have been widely used due to their simplicity [16], [17]. Markov Models and Hidden Markov Models (HMM) [18] are also used to model the transition between instances to describe sequences. Their training procedure can be either generative or discriminative [19]. In the first case, each class is represented by a model which describes the class properties, while in the later, the training process takes into account the discriminant information among the classes, in order to increase the classification performance. The aforementioned techniques assume independence between sequence instances, which is often not realistic. Conditional Random Fields (CRF) [20] relax this independence assumption, while increasing classification accuracy.

Feature-based classification methods represent each sequence with one or more features, which are subsequently used for classification. n-gram based sequence representation can be achieved by applying discretization on the sequence instances, or by performing clustering and representing each instance with the closest cluster centroid. Another approach is to consider sequences as shapes and describe

their properties. Shapelets [21] are defined as the subsequences capturing local sequence properties. A sequence can be described as a collection of shapelets. Wavelet decomposition has been applied to sequence classification, in order to capture both local and global sequence properties [22]. Specifically, the low-order wavelet coefficients have been used to describe global sequence properties, while high-order coefficients have been used to describe the local sequence information. After finding a convenient sequence representation, standard classification techniques, such as Support Vector Machines (SVM), K-nearest neighbors or Artificial Neural Networks (ANN) can be utilized for sequence classification.

In this paper, we propose a novel multidimensional sequence classification method. We take into account the general case of multidimensional real-valued instances. The training sequence instances $\mathbf{s}_n \in \mathbb{R}^N$ are clustered in order to produce K instance prototypes, i.e., in order to determine an instance codebook. The labeling information available for the training sequences is, subsequently, exploited to tune the instance codebook and increase sequence class discrimination, thus determining a set of K optimal instance representations, which are called *statemes* (stemming from the word state). The statemes define the so-called stateme space $\mathbb{R}^K$, as follows. After statemes determination, a sequence is represented in a new way, by a vector denoting the similarity between its instances and all chosen statemes. This is a convenient sequence representation that can be applied to any sequence classification problem. For example, in handwritten character or human action recognition problems, the sequence representation is in the form of a real-valued vector denoting the similarity of the test sequence (representing one character or action, respectively) with the corresponding statemes. Subsequently, the class label information that is available for the training sequences, is exploited in order to determine an optimal linear transformation of the stateme space $\mathbb{R}^K$ to a low-dimensional feature space $\mathbb{R}^D$, $D \ll K$, which enhances class discrimination. This is achieved by exploiting, once again, the available labels of the training sequences, in order to determine both the optimal linear transform matrix and the optimal sequence class representation in the stateme space, in terms of sequence class discrimination. The final sequence classification in this space can be performed by various methods, such as the nearest class centroid(s) [23], providing high classification accuracy. In

this paper we perform classification based on a modified nearest subclass centroid scheme. The proposed classification procedure has low computational cost and requires small amount of memory, since each sequence class is represented by few low-dimensional feature vectors, since $D \ll K \ll N$. Furthermore, by allowing multiple subclasses in each sequence class, we take into account intra-class structure variations, which are usual in real applications. For example in handwritten character classification each person writes in a different way that slightly varies from time to time, thus creating subclasses within each handwritten character class.

Vector quantization based data representation has been widely used in image/video analysis [24], [25]. In such data representations, representative feature vectors are extracted from the training images/videos to form a so-called codebook. Based on this codebook, data are represented by performing hard or soft feature vector quantization. It should be noted that most image/video analysis methods employ standard clustering techniques, like K-Means [6], for codebook construction. That is, the codebook is obtained in an unsupervised manner. Supervised codebook construction is a, relatively, new task [26], [27], [28], [29], [30], [31]. The main idea behind supervised codebook construction is the incorporation of the labeling information, that is available for the training data, in order to construct better clusters, in terms of intrinsic cluster structure. A probabilistic codebook learning technique is proposed in [26], [27], [28]. According to this, class-specific codebooks are learned from an initial codebook, by adapting or merging codewords appearing in an initial universal codebook. The information bottleneck principle, combined with kernel density estimation, has been exploited for codebook learning in [29]. Extremely Randomized Clustering Forests (ERC-Forests) have been propose in [30], where the given class labels are exploited in order to control the codebook size. A codebook learning method based on minimum between-category mutual information loss is proposed in [31]. In all the above described cases, the adoption of the corresponding supervised codebook has led to an increase of the classification performance, compared to the unsupervised case. However, the supervised codebook construction process is not directly related to the discrimination of the classes involved in the classification problem. In this paper, we aim at determining the optimal

codebook (statemes), in terms of better sequence classes discrimination. To this end, we propose a novel codebook learning method, based on the minimization of the within-class scatter to the between-class scatter ratio. This optimization criterion is evaluated on sequential data, by adopting the mean fuzzy distances of a stateme based sequence representation.

Dimensionality reduction techniques have been widely used in classification schemes due to their ability to reduce the data dimensionality and enhance class discrimination. Linear Discriminant Analysis (LDA) [6] is, probably, the most widely adopted technique. The main idea of standard LDA is to find an optimal reduced-dimensionality space for data projection, in which the classes are better discriminated. The adopted criterion is the ratio of the within-class scatter to the between-class scatter in the projection space. By minimizing this criterion, maximal class discrimination is achieved. Clustering Discriminant Analysis (CDA) [32] is a generalization of LDA that takes into account subclass information. Standard LDA and CDA techniques try to determine the optimal data projection matrix, by representing classes using the corresponding mean (sub)class vectors. However, other, optimized, (sub)class vector representations can be found that provide better class discrimination. In this paper, we relax the assumption of class representation by the corresponding mean (sub)class vectors and we propose an iterative optimization scheme aiming at determining both the optimal, in terms of class discrimination, representative (sub)class vectors and the the data projection matrix for CDA based projection.

Overall, the contributions of this paper are: $1)$ the proposition of a novel iterative optimization procedure, which provides both the optimal representative sequence (sub)class vectors and the optimal projection matrix for CDA-based dimensionality reduction, $2)$ the proposition of a novel iterative optimization procedure, in order to increase instance codebook discrimination power, based on CDA criterion minimization, $3)$ the proposition of a unified framework for multidimensional sequence classification involving three optimization steps optimizing the same sequence class discrimination criterion.

The remainder of this paper is structured as follows. Section II presents the steps performed by the proposed method. Section III presents experiments conducted on publicly available databases coming from

a wide variety of sequence classification problems in order to assess the performance of the proposed method. Finally, conclusions are drawn in Section V.

In terms of notation, we use superscripts to denote sequence class and sequence subclass indices and subscripts to denote sequence number and instance number indices. For example, we use the notation $\mathbf{s}_{nm}^{ij}$ to denote the m-th instance of sequence n, belonging to subclass j of sequence class i. In cases where the sequence class information is not necessary, we drop superscipts. That is, we use the notation $\mathbf{s}_{nm}$ to denote instance m of the n-th training sequence.

II. Proposed Method

In this paper, a sequence of multidimensional instances is denoted by $\mathbf{s}_n = [\mathbf{s}_{n1}, \mathbf{s}_{n2}, ..., \mathbf{s}_{nN_n}]^T$, where $\mathbf{s}_{nm} \in \mathbb{R}^N$ is the m-th instance of sequence n, N_n is the number of instances forming sequence $\mathbf{s}_n$ and may vary over sequences. Let $\mathcal{L} = \{L_1, ..., L_M\}$ be a set of sequence classes, e.g. 'walk', 'run', 'jump', etc, in the case of human actions. Given a set $\mathcal{S}$ of N_T training sequences $\mathcal{S} = \{\mathcal{S}_1, ..., \mathcal{S}_{N_T}\}$, where $\mathcal{S}_i = \{\mathbf{s}_1^i, ..., \mathbf{s}_{N^i}^i\}$ is a subset of N^i sequences belonging to sequence class L_i, we would like to devise a sequence classification algorithm which addresses all the previously denoted challenges by finding an optimal sequence representation in terms of statemes and dimensionality reduction during training. After training, a test procedure can be applied to a test sequence $\mathbf{s}_{test}$ in order to find its class label l_{test}. In the following, we present in detail each individual step of the proposed method.

A. Statemes Determination

In the training phase, all training sequence instances $\mathbf{s}_{nm}$ are clustered, in order to produce K statemes, $\mathbf{v}_k$, $k = 1, ..., K$, without taking into account the known training sequence instance labels. Several clustering algorithms can be utilized for this task. We have conducted experiments using K-Means [33], Fuzzy C-Means [34], Self Organizing Maps [35] and Spectral Clustering [36], [37] and have experimentally found that the choice of the clustering algorithm does not affect significantly the performance of the proposed method. As K-Means is a fast clustering algorithm, we chose it to cluster the training instances

by minimizing $\sum_{k=1}^{K}\sum_{n=1}^{N_T}\sum_{m=1}^{N_n} a_{nmk} \parallel \mathbf{s}_{nm} - \mathbf{v}_k \parallel^2$, where N_n is the number of instances forming sequence $\mathbf{s}_n$. The parameter a_{nmk} is chosen as follows: $a_{nmk} = 1$ if instance $\mathbf{s}_{nm}$ is assigned to the k-th cluster (having cardinality $n_k = \sum a_{nmk}$) and $a_{nmk} = 0$ otherwise. The K statemes $\mathbf{v}_k$ are obtained by calculating the arithmetic mean of the instances assigned to each of these K clusters:

$$\mathbf{v}_k = \frac{1}{n_k}\sum_{n=1}^{N_T}\sum_{m=1}^{N_n} a_{nmk}\mathbf{s}_{nm}. \tag{1}$$

The optimal number of statemes is determined by performing multiple fold (typically 10-fold) cross-validation procedure. The training set is divided in ten subsets and, in each fold of the cross-validation procedure, the algorithm is trained using nine subsets and tested using the remaining one. An experiment consists of ten folds, one for each test subset.

In the following section, we propose a stateme-based sequence representation. Specifically, we propose a non-linear mapping of a sequence to a feature space determined by the statemes, called stateme space. In order to enhance the sequence class discrimination in the statemes space, we propose an iterative procedure in order to determine the optimal, in terms of classification accuracy, stateme choice. This procedure is described in Subsection II-C.

B. Sequence Representation

After stateme calculation, each sequence $\mathbf{s}_n$ is mapped to the so-called *sequence vector* $\mathbf{q}_n \in \mathbb{R}^K$, which denotes the similarity of the instances forming the sequence $\mathbf{s}_{nm}$ to all statemes $\mathbf{v}_k,\ k = 1, ..., K$, based on the fuzzy distances between the instances $\mathbf{s}_{nm}$ an the dynemes $\mathbf{v}_k$:

$$d_{nmk} = (\parallel \mathbf{s}_{nm} - \mathbf{v}_k \parallel_2)^{-\frac{2}{r-1}}. \tag{2}$$

r is the fuzzification parameter ($r > 1$), which is also determined by applying the cross-validation procedure. We have experimentally found that the value $r = 1.1$ provides a satisfactory sequence representation. Fuzzy distances allow smooth distance representation between sequence instances and statemes. Fuzzy distances d_{nmk} are used to form the so-called *stateme distance vector* $\mathbf{d}_{nm} = [d_{nm1}, d_{nm2}, ..., d_{nmK}]^T \in \mathbb{R}^K$,

Stateme distance vectors $\mathbf{d}_{nm}$ are normalized to produce the instance membership vectors $\mathbf{u}_{nm} \in \mathbb{R}^K$, which are the instances $\mathbf{s}_{nm}$ representation in the stateme space $\mathbf{u}_{nm} = \frac{\mathbf{d}_{nm}}{\|\mathbf{d}_{nm}\|}$. Sequence vector $\mathbf{q}_n$ are determined as the mean membership vector $\mathbf{q}_n = \frac{1}{N_n} \sum_{m=1}^{N_n} \mathbf{u}_{nm}$.

By using a simple arithmetic mean for sequence representation, we avoid taking into account any instance order, (i.e. temporal) information in the sequence representation, hence efficiently handling, e.g. variations in sequence length. Finally, the training sequence vectors $\mathbf{q}_n$ are normalized to have zero mean and unit standard deviation, producing the normalized sequence vectors $\mathbf{x}_n$, which represent sequences $\mathbf{s}_n$ in the stateme space. Since our final purpose is to obtain the optimal, in terms of classification accuracy, sequence representation, we are interested to enhance the discriminant ability of $\mathbf{x}_n$. In order to achieve that, we can a) optimize the choice of statemes $\mathbf{v}_k$ and b) map $\mathbf{x}_n$ to a more discriminant sequence representation $\mathbf{z}_n$ in a reduced-dimensionality space $\mathbb{R}^D, D \ll K$. In this paper we follow both these directions, as described in the subsequent sections.

C. Stateme Optimization

In this section, we present an iterative optimization procedure aiming to determine the optimal statemes $\mathbf{v}_k$, in terms of sequence class discrimination. Since $\mathbf{x}_n$ are functions of $\mathbf{v}_k$, the optimal sequence representation in the stateme space can be obtained by minimizing Fisher [6] sequence class discrimination criterion with respect to $\mathbf{v}_k$:

$$\mathcal{J}_1 = \frac{trace(\mathbf{S}_w)}{trace(\mathbf{S}_b)}, \tag{3}$$

where $\mathbf{S}_w$, $\mathbf{S}_b$ denote the within-class and between-class scatter matrices, respectively. Fisher criterion is typically used for unimodal classes. When multimodality exists, i.e., each class consists of several subclasses, CDA performs better. Let us assume that the sequence class i consists of c_i subclasses, each containing N^{ij} training normalized sequence vectors and represented by the corresponding mean vector $\boldsymbol{\mu}^{ij}$. Let us denote by $\mathbf{x}_n^{ij}$ the n-th training normalized sequence vector belonging to the j-th subclass of

sequence class i. In CDA, $\mathbf{S}_w$ and $\mathbf{S}_b$ are defined as follows:

$$\mathbf{S}_w = \sum_{i=1}^{M}\sum_{j=1}^{c_i}\sum_{n=1}^{N^{ij}}(\mathbf{x}_n^{ij} - \boldsymbol{\mu}^{ij})(\mathbf{x}_n^{ij} - \boldsymbol{\mu}^{ij})^T \tag{4}$$

$$\mathbf{S}_b = \sum_{i=1}^{M}\sum_{l\neq i}\sum_{j=1}^{c_i}\sum_{h=1}^{c_l}(\boldsymbol{\mu}^{ij} - \boldsymbol{\mu}^{lh})(\boldsymbol{\mu}^{ij} - \boldsymbol{\mu}^{lh})^T \tag{5}$$

The minimization of $\mathcal{J}_1$, can be done by a gradient descent procedure:

$$\mathbf{v}_k(t) = \mathbf{v}_k(t-1) - \alpha\frac{\partial\mathcal{J}_1(t-1)}{\partial\mathbf{v}_k(t-1)}, \tag{6}$$

where t denotes the iteration of the update procedure and α is an update rate parameter. In our experiments we used the value $\alpha = 0.5$.

$\frac{\partial\mathcal{J}}{\partial\mathbf{v}_k}$ takes the form:

$$\begin{aligned}
\frac{\partial\mathcal{J}_1}{\partial\mathbf{v}_k} &= \frac{2}{trace(\mathbf{S}_b)}(\mathbf{x}_n^{ij} - \boldsymbol{\mu}^{ij}) - \frac{2trace(\mathbf{S}_w)}{N^{ij}trace(\mathbf{S}_b)^2}\sum_{l\neq i}\sum_{h=1}^{c_l}(\boldsymbol{\mu}^{ij} - \boldsymbol{\mu}^{lh}) \\
&\times e_k \\
&\times \frac{1}{\tilde{q}_k}\left(1 - \frac{1}{N_T}\right) - \frac{q_{nk}^{ij} - \bar{q}_k}{\tilde{q}_k^3(N_T-1)}\left[(q_{nk}^{ij} - \bar{q}_k) - \frac{1}{N_T}\sum_{l=1}^{M}\sum_{h=1}^{c_l}\sum_{m=1}^{N^{lh}}(q_{mk}^{lh} - \bar{q}_k)\right] \\
&\times \frac{1}{N_n^{ij}}\sum_{m=1}^{N_n^{ij}}\left\{\frac{1}{\left[\sum_{l=1}^{K}(d_{nml}^{ij})^2\right]^{1/2}} - \frac{(d_{nmk}^{ij})^2}{\left[\sum_{l=1}^{K}(d_{nml}^{ij})^2\right]^{3/2}}\right\} \\
&\times \frac{-g(\mathbf{v}_k - \mathbf{s}_m)}{\|\mathbf{v}_k - \mathbf{s}_{nm}^{ij}\|_2^{g+2}}
\end{aligned} \tag{7}$$

where N_n^{ij} is the number of instances forming $\mathbf{s}_n^{ij}$, $e_k \in \mathbb{R}^K$ ($e_i = 0$ for $i \neq k$ and $e_i = 1$ for $i = k$), $\bar{q}_k = \frac{1}{N_T}\sum_{i=1}^{M}\sum_{j=1}^{c_i}\sum_{n=1}^{N^{ij}} q_{nk}^{ij}$ is the k-th element of the mean training sequence vector, $\tilde{q}_k = \frac{1}{N_T-1}\sum_{i=1}^{M}\sum_{j=1}^{c_i}\sum_{n=1}^{N^{ij}}(q_{nk}^{ij} - \bar{q}_k)$ is the corresponding standard deviation and $g = \frac{2}{r-1}$.

In the training phase, we initialize $\mathbf{v}_k$ by clustering all the instances forming the training sequences, as described in section II-A. Based on these statemes $\mathbf{v}_k(0)$, the training sequences are mapped to the statemes space to obtain $\mathbf{x}_n^{ij}(0)$. In this space, normalized sequence vectors $\mathbf{x}_n^{ij}(0)$ belonging to each sequence class are clustered, in order to obtain the sequence class centers $\boldsymbol{\mu}^{ij}(0)$. Scatter matrices $\mathbf{S}_w(0)$, $\mathbf{S}_b(0)$ are calculated by using (4), (5) for $\mathbf{x}_n^{ij}(0)$ and $\boldsymbol{\mu}^{ij}(0)$. The criterion value $\mathcal{J}_1(0)$ is, subsequently, calculated via (3).

The iterative procedure is performed multiple times, by introducing the training sequences in a random order. At iteration t, $\mathbf{v}_k(t)$ are calculated using (6). Based on $\mathbf{v}_k(t)$, new $\mathbf{x}_n^{ij}(t)$ are calculated, which are assigned to subclasses based on their distances from the subclasses centers $\boldsymbol{\mu}^{ij}(t-1)$, using the Euclidean distance. New subclass centers $\boldsymbol{\mu}^{ij}(t)$ are, subsequently, calculated as the mean subclass vectors in the stateme space. Scatter matrices $\mathbf{S}_w(t)$ and $\mathbf{S}_b(t)$ are, finally, computed using (4), (5), resulting to a new criterion value $\mathcal{J}_1(t)$. This procedure is performed until the criterion $\mathcal{J}_1(t+1) - \mathcal{J}_1(t) < \epsilon_1$, where ϵ_1 is a small positive value, or for a fixed number of iterations.

D. CDA Projection

After finding the optimal sequence representation in the stateme space, the labeling information available in the training phase can be exploited, in order to determine an optimal discriminant space, where the projected sequence vectors belonging to different classes are better separated. We employ CDA to this end looking for a linear transform $\boldsymbol{\Psi}_{opt}$, which maps the normalized sequence vectors $\mathbf{x}_n^{ij}$ to discriminant sequence vectors $\mathbf{z}_n^{ij} \in \mathbb{R}^D$ by applying $\mathbf{z}_n^{ij} = \boldsymbol{\Psi}_{opt}^T \mathbf{x}_n^{ij}$. The dimensionality of the final sequence representation is at most $D = C - 1$, where C is the total number of subclasses forming the sequence classes: $C = \sum_{i=1}^{M} c_i$. $\boldsymbol{\Psi}_{opt}$ is determined by minimizing the criterion, $\boldsymbol{\Psi}_{opt} = \arg\min_{\boldsymbol{\Psi}}\{\mathcal{J}_2\}$:

$$\mathcal{J}_2 = \frac{trace\{\boldsymbol{\Psi}^T \mathbf{S}_w \boldsymbol{\Psi}\}}{trace\{\boldsymbol{\Psi}^T \mathbf{S}_b \boldsymbol{\Psi}\}}, \tag{8}$$

where $\mathbf{S}_w$, $\mathbf{S}_b$ are given by (4) and (5), respectively. The optimization problem in (8) is equivalent to the optimization problem $\mathbf{S}_w \mathbf{v} = \lambda \mathbf{S}_b \mathbf{v}$, $\lambda \neq 0$, which can be solved by performing eigenanalysis to the matrix $\mathbf{S}_b^{-1}\mathbf{S}_w$. The optimal projection matrix $\boldsymbol{\Psi}$ is formed by the eigenvectors corresponding to the D largest eigenvalues, since the remaining eigenvalues are equal to zero.

The standard CDA algorithm described above provides an optimal discriminant space assuming that each subclass is represented by the corresponding mean vector $\boldsymbol{\mu}^{ij}$. However, other 'center' vectors $\tilde{\boldsymbol{\mu}}^{ij}$ can be found to represent the subclasses by an optimization procedure, minimizing $\mathcal{J}_2$, which is a function of $\tilde{\boldsymbol{\mu}}^{ij}$ according to (4), (5), (8). That is, subclass representation by different 'center' vectors $\tilde{\boldsymbol{\mu}}^{ij}$ results

in the derivation of different 'scatter' matrices $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$, by using $\tilde{\boldsymbol{\mu}}^{ij}$ instead of $\boldsymbol{\mu}^{ij}$ in (4), (5). They in turn, result in the determination of a different projection matrix $\tilde{\boldsymbol{\Psi}}_{opt}$, thus determining a different projection subspace.

In order to determine the optimal $\tilde{\boldsymbol{\mu}}^{ij}$, we perform an iterative procedure based on the steepest descend approach. This procedure starts by using the solution given by the standard CDA algorithm, i.e., $\tilde{\boldsymbol{\mu}}^{ij}(0) = \boldsymbol{\mu}^{ij}$ and $\tilde{\mathbf{S}}_w(0) = \mathbf{S}_w$, $\tilde{\mathbf{S}}_b(0) = \mathbf{S}_b$. By calculating $\frac{\partial \mathcal{J}_2}{\partial \tilde{\boldsymbol{\mu}}^{ij}}$ we obtain:

$$\frac{\partial \mathcal{J}_2}{\partial \tilde{\boldsymbol{\mu}}^{ij}} = \frac{\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T \left[\frac{2}{N^{ij}} \sum_{n=1}^{N_{ij}} \left(\tilde{\boldsymbol{\mu}}^{ij} - \mathbf{x}_n^{ij}\right)\right]}{trace(\tilde{\boldsymbol{\Psi}}^T \tilde{\mathbf{S}}_b \tilde{\boldsymbol{\Psi}})} - \frac{trace(\tilde{\boldsymbol{\Psi}}^T \tilde{\mathbf{S}}_w \tilde{\boldsymbol{\Psi}}) \left[\tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{\Psi}}^T \left(\frac{2}{C} \sum_{l \neq i} \sum_{h=1}^{c_l} \left(\tilde{\boldsymbol{\mu}}^{ij} - \tilde{\boldsymbol{\mu}}^{lh}\right)\right)\right]}{trace(\tilde{\boldsymbol{\Psi}}^T \tilde{\mathbf{S}}_b \tilde{\boldsymbol{\Psi}})^2}, \quad (9)$$

$$\tilde{\boldsymbol{\mu}}^{ij}(t) = \tilde{\boldsymbol{\mu}}^{ij}(t-1) - \beta \frac{\partial \mathcal{J}_2(t-1)}{\partial \tilde{\boldsymbol{\mu}}^{ij}(t-1)}. \quad (10)$$

where β is an update rate parameter. In our experiments we used the value $\beta = 0.1$.

After calculating $\tilde{\boldsymbol{\mu}}^{ij}(t)$, the scatter matrices $\tilde{\mathbf{S}}_w(t)$ and $\tilde{\mathbf{S}}_b(t)$ given in equations (4) and (5), respectively, are calculated and a new optimal projection matrix $\tilde{\boldsymbol{\Psi}}_{opt}(t)$ is determined. The normalized sequence vectors $\mathbf{x}_n^{ij}$ are mapped to the space specified by $\tilde{\boldsymbol{\Psi}}_{opt}(t)$ and the criterion value $\mathcal{J}_2(t+1)$ is calculated. This procedure is performed until $\mathcal{J}_2(t+1) - \mathcal{J}_2 t < \epsilon_2$, where ϵ_2 is a small positive value, or for a fixed number of iterations.

E. Optimal sequence class representation in the decision space

After determining $\tilde{\boldsymbol{\Psi}}_{opt}$, normalized sequence vectors $\mathbf{x}_n^{ij}$ are mapped to the discriminant sequence vectors $\tilde{\mathbf{z}}_n^{ij}$, by applying $\tilde{\mathbf{z}}_n^{ij} = \tilde{\boldsymbol{\Psi}}_{opt}^T \mathbf{x}_n^{ij}$. In this space, sequence classification can be performed by using several classification methods, such as Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), Nearest Neighbors (K-NN), Nearest subclass centroid, etc. In this paper, we employ Nearest subclass centroid for sequence classification. That is, each sequence class i is represented by c_i vectors, $\mathbf{m}^{ij} \in \mathbb{R}^D$, $i = 1, ..., M$, $j = 1, ..., c_i$. Given $\mathbf{m}^{ij}$, a test discriminant sequence vector $\tilde{\mathbf{z}}_{test}$, can be classified to the sequence class of the closest subclass vector $\mathbf{m}^{ij}$, by using the Euclidean distance:

$$l_{test} = \arg\min_i \|\mathbf{m}^{ij} - \tilde{\mathbf{z}}_{test}\|_2, \;\; i = 1, ..., M, \; j = 1, ..., c_i. \quad (11)$$

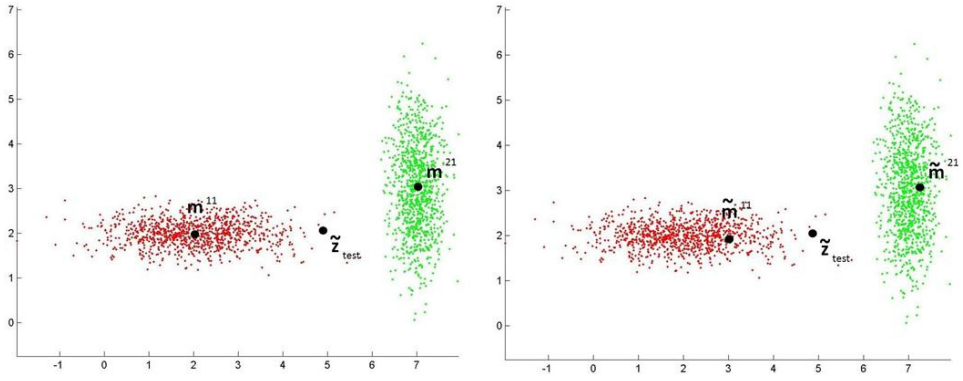

Fig. 4. *a) Class representation using the mean class vectors* $\mathbf{m}^{ij}$ *and b) Class representation using* $\tilde{\mathbf{z}}_{test}$ *resulting to higher classification accuracy.*

Usually, $\mathbf{m}^{ij}$ are determined to be the mean subclass discriminant sequence vectors, i.e., $\mathbf{m}^{ij} = \frac{1}{N^{ij}} \sum_{n=1}^{N^{ij}} \mathbf{x}_n^{ij}$. In our case, we employ the optimal subclass representation in the statemes space $\tilde{\boldsymbol{\mu}}^{ij}$ determined in section II-D for $\mathbf{m}^{ij}$ calculation $\mathbf{m}^{ij} = \tilde{\boldsymbol{\Psi}}_{opt}^T \tilde{\boldsymbol{\mu}}^{ij}$.

However, this may not be the optimal choice for class representation. Consider the simple example illustrated in Figure 4, where we present two classes, each consisting of one subclass. We assume that classification is performed using the nearest class centroid and the Euclidean distance. By representing each class with the corresponding mean class vector $\mathbf{m}^{ij}$, shown in Figure 4a, $\tilde{\mathbf{z}}_{test}$ will falsely be assigned to class 2, whereas by representing subclasses with vectors $\tilde{\mathbf{m}}^{ij}$, shown in Figure 4b, $\tilde{\mathbf{z}}_{test}$ will be correctly classified to class 1.

Motivated by this observation, we would like to determine the optimal $\tilde{\mathbf{m}}^{ij}$ in terms of class discrimination. This can be done by minimizing the CDA criterion, measuring class compactness, with respect to $\tilde{\mathbf{m}}^{ij}$:

$$\mathcal{J}_3 = \frac{trace(\hat{\mathbf{S}}_w)}{trace(\hat{\mathbf{S}}_b)}. \tag{12}$$

$\hat{\mathbf{S}}_w$ and $\hat{\mathbf{S}}_b$ are the within-class and between-class scatter matrices in the decision space, defined as:

$$\hat{\mathbf{S}}_w = \sum_{i=1}^{M} \sum_{j=1}^{c_i} \sum_{n=1}^{N^{ij}} (\tilde{\mathbf{z}}_n^{ij} - \tilde{\mathbf{m}}^{ij})(\tilde{\mathbf{z}}_n^{ij} - \tilde{\mathbf{m}}^{ij})^T \tag{13}$$

$$\hat{\mathbf{S}}_b = \sum_{i=1}^{M} \sum_{l \neq i} \sum_{j=1}^{c_i} \sum_{h=1}^{c_l} (\tilde{\mathbf{m}}^{ij} - \tilde{\mathbf{m}}^{lh})(\tilde{\mathbf{m}}^{ij} - \tilde{\mathbf{m}}^{lh})^T \tag{14}$$

By calculating $\frac{\partial \mathcal{J}_3}{\partial \tilde{\mathbf{m}}^{ij}}$ we obtain:

$$\frac{\partial \mathcal{J}_3}{\partial \tilde{\mathbf{m}}^{ij}} = \frac{2\sum_{n=1}^{N^{ij}}(\tilde{\mathbf{m}}^{ij} - \mathbf{z}_n^{ij})}{trace(\hat{\mathbf{S}}_b)} - \frac{2trace(\hat{\mathbf{S}}_w)\sum_{l\neq i}\sum_{h=1}^{c_l}\left(\tilde{\mathbf{m}}^{ij} - \tilde{\mathbf{m}}^{lh}\right)}{trace(\hat{\mathbf{S}}_b)^2}, \tag{15}$$

$$\tilde{\mathbf{m}}^{ij}(t) = \tilde{\mathbf{m}}^{ij}(t-1) - \gamma\frac{\partial \mathcal{J}_3(t-1)}{\partial \tilde{\mathbf{m}}^{ij}(t-1)}. \tag{16}$$

γ is an update rate parameter. In our experiments we used the value $\gamma = 0.1$.

$\tilde{\mathbf{m}}^{ij}$ are initialized to the vectors $\mathbf{m}^{ij}$, i.e., $\tilde{\mathbf{m}}^{ij}(0) = \mathbf{m}^{ij}$. After calculating $\tilde{\mathbf{m}}^{ij}(t)$, the scatter matrices $\hat{\mathbf{S}}_w(t)$ and $\hat{\mathbf{S}}_b(t)$, (13), (14), respectively, are calculated resulting to a new criterion value $\mathcal{J}_3(t)$. This procedure is performed until $\mathcal{J}_3(t+1) - \mathcal{J}_3(t) < \epsilon_3$, where ϵ_3 is a small positive value, or for a fixed number of iterations.

The above described procedure can be performed to any classification task following the nearest subclass classification approach. However, we should note that, in our case, it is equivalent with the optimization procedure described in section II-D, for fixed projection matrix $\tilde{\mathbf{\Psi}}_{opt}$. That is, after determining the optimal discriminant subspace, obtained by the corresponding projection matrix $\tilde{\mathbf{\Psi}}_{opt}$, we further modify the subclasses centers in the decision space for better sequence class discrimination.

F. Sequence Classification

In order to classify a novel (test) sequence $\mathbf{s}_{test}$, it is mapped to the stateme space, as described in section II-C. The test normalized sequence vector $\mathbf{x}_{test}$ is, subsequently, projected to the discriminant subspace determined by $\tilde{\mathbf{\Psi}}_{opt}$ to obtain the discriminant sequence vector $\tilde{\mathbf{z}}_{test} = \tilde{\mathbf{\Psi}}_{opt}^T\mathbf{x}_{test}$. Finally, $\tilde{\mathbf{z}}_{test}$ is assigned to the sequence class label corresponding to the nearest subclass centroid $\tilde{\mathbf{m}}^{ij}$, by using the Euclidean distance, i.e.:

$$l_{test} = \arg\min_i \|\tilde{\mathbf{m}}^{ij} - \tilde{\mathbf{z}}_{test}\|_2, \;\; i = 1, ..., M, \; j = 1, ..., c_i. \tag{17}$$

III. Experimental Results

In this section, we present experiments conducted in order to evaluate the proposed method. We have used publicly available databases coming from a wide range of applications, in order to assess

its effectiveness in different sequence classification problems. The used databases come from handwritten character recognition, time series classification and human action recognition.

In all our experiments, sequences $\mathbf{s}_n$, consisting of instances $\mathbf{s}_{nm} \in \mathbb{R}^N$, were mapped to sequences $\tilde{\mathbf{s}}_n$ having instances taking values in $[0, 1]$, i.e., $\tilde{s}_{nml} = \frac{s_{nml} - \min\limits_{l}(s_{nml})}{\max\limits_{l}(s_{nml}) - \min\limits_{l}(s_{nml})},\ l = 1, ..., N$. By using $\tilde{\mathbf{s}}_n$ instead of $\mathbf{s}_n$, we address issues related to sequence displacements and scaling that may appear in the original sequence representation.

A. Experiments on Handwritten Character Recognition

We have used a publicly available handwritten character recognition database [38], [39] consisting of 2858 character samples, divided in 20 classes. Data were captured using a WACOM tablet. They have three dimensions: pen tip coordinates $[x, y]^T$ and pen tip force f. The sampling frequency was equal to 200Hz. The data have been numerically differentiated and smoothed using a Gaussian filter with a sigma value equal to 2. Sample sequences are illustrated in Figure 5a.

For the first set of experiments, we assume that each character is a sequence of 2D contour pixel coordinates $[x, y]^T$ forming the character. As was expected, the number of contour points of each character in the database may vary. Multiple experiments have been performed, for different numbers of statemes and subclasses per class. In order to compare the performance of the proposed method with that of other methods proposed in the literature, we used the first twenty characters per class for testing and the remaining ones for training. A classification accuracy equal to 90% was obtained when using 25 statemes and two subclasses per class. Another set of experiments has been performed by representing each character as a sequence of 3-dimensional vectors $[x, y, f]^T$. This resulted in better character representation and an increase in classification accuracy. Figure 5 illustrates the classification rates obtained for the second set of experiments. As can be seen, by using 5 statemes and one subclass per class a sequence classification accuracy equal to 65% has been obtained. The corresponding classification accuracy for the cases of two and three subclasses per class are equal to 70% and 75%, respectively. This is reasonable, as the data multimodality is better addressed by using a higher number of subclasses. By increasing the number

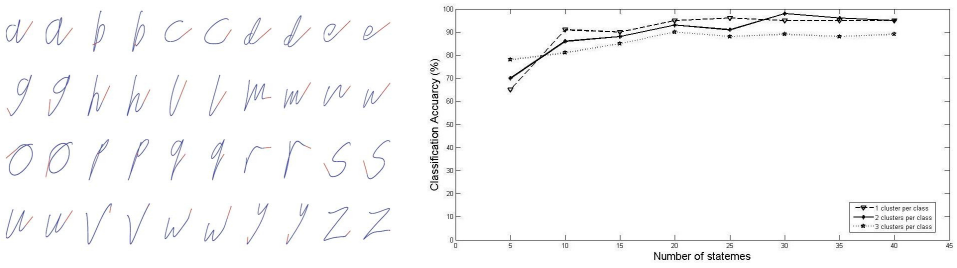

Fig. 5. *a) Sample sequences from the handwritten character recognition database and b) character recognition using different number of statemes and subclasses per class.*

of statemes, sequence representation in the statemes space becomes more discriminant, resulting to an increase of the classification accuracy. The obtained classification accuracies indicate that 10 statemes are sufficient for good character representation, providing classification accuracies equal to 91%, 86% and 81% for one, two and three subclasses per class, respectively. The best classification accuracy, equal to 98.25% has been obtained by using 30 statemes and two subclasses per class. Overall, the number of statemes and the number of subclasses forming each sequence class play a significant role in the performance of the proposed method. When using few statemes, the sequence representation in the stateme space is inaccurate. In this case, the use of multiple subclasses per sequence class will probably increase the performance of the proposed method. By using a high number of statemes, sequences are described in more detail and, thus, the final sequence representation is more discriminant. In such cases, the use of many subclasses per sequence class does not help. However, too many statemes result in an overcomplete representation that does not increase the discriminant power any further. In Table Ib we compare the performance of our method with that of other methods proposed in the literature, in this handwritten character database. As can be seen, the proposed approach outperforms existing sequence classification methods in this experimental setting, providing very satisfactory classification accuracy.

B. Experiments in Time Series Classification

The UCR time series database [42] includes 6 two-class and 14 multi-class sequence classification problems for a wide variety of applications. These include biomedical data classification, electromagnetic

TABLE I

COMPARISON RESULTS IN THE HANDWRITTEN CHARACTER RECOGNITION DATABASE.

Method	**Accuracy**
Jaakkoda and Haussler [40]	89.26%
Perina et.al. [4]	92.91%
Tsuda et.al. [41]	93.67%
Proposed method	**98.25%**

measurements, synthetic data, etc. For each data set, a training and a test set are provided in the database. Data forming each data set are sequences of real-valued instances. Information about the data sets can be found in Table II. Classification results for several algorithms, including K nearest neighbors (KNN), Multi-Layer Perceptron (MLP) and Support Vector Machines (SVM) are already available for this database.

In our experiments, we represented each sequence instance as a 2D vector having the form $\mathbf{p}_{in} = [n, v_n]^T$, where n refers to the instance order in the sequence and v_n refers to the observed value. This is a commonly used time series representation [43], [44]. In order to evaluate the contribution of each optimization step to the efficiency of the proposed method, we conducted experiments by using three variants of the proposed algorithm. In the first one, no optimization procedure has been performed. That is, we used the statemes calculated by clustering the training sequence instances and we performed CDA by using the mean subclass vectors. In the second one, only the stateme optimization procedure has been performed, while in the third one, all three optimization procedures have been performed. Multiple experiments have been performed in order to determine the optimal parameters for all variants described above. Figure 6 illustrates an example of the three criteria optimization procedures obtained for the 'fish' data set, using 36 statemes and two subclasses per sequence class. In Figure 6a we present the sequence instances forming all the training sequences in black color. Statemes calculated by clustering the training instances are illustrated in blue color, while the optimized statemes are illustrated in red color. Figures 6b, 6c and 6d illustrate the $\mathcal{J}_1$, $\mathcal{J}_2$ and $\mathcal{J}_3$ values obtained during the stateme, CDA and subclass centers

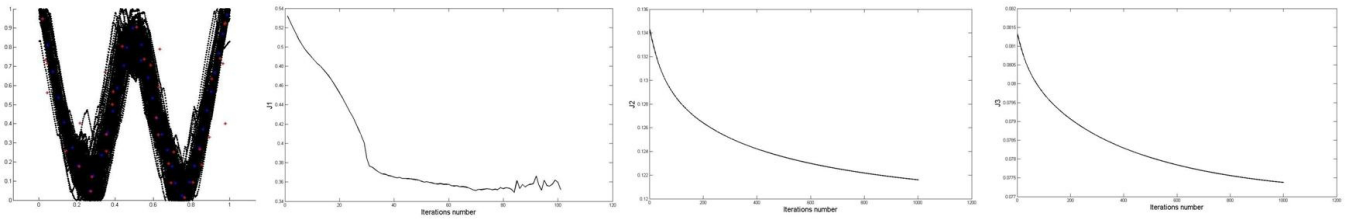

Fig. 6. *a) Sequence instances forming the training sequences (black), initial (blue) and optimized (red) statemes. b) J*1*, c) J*2*, and c) J*3 *optimization criteria values.*

optimization procedures, respectively.

The classification rates obtained by using the optimal parameters for each data set are given in Table II. As can be seen, each optimization step contributes to the increase of the performance of the proposed method. For all data sets, the best classification rates have been invariably obtained by employing all three optimization procedures. In Table II, we also present classification results for other classification algorithms applied to the same data sets. As can be seen, there is no unique best algorithm for all sequence classification problems included in the database. MLP provides the highest classification accuracy in 4 out of the 20 data sets, while SVMs and KNN are best in 3 out of 20 data sets. The proposed method is clearly the overall winner, since it outperforms all other methods in 9 out of 20 data sets.

C. Experiments on Human Action Recognition

In order to illustrate the effectiveness of the proposed method in multidimensional sequence classification, we conducted experiments on the i3DPost 8-view action recognition database [45]. This database contains 64 high-resolution (1920×1080 pixel) image sequences of eight persons (six males and two females), each performing eight actions: {'walk', 'run', 'jump1', 'jump2', 'bend', 'fall', 'sit', 'wave'}. Actions were captured from eight cameras placed around a capture volume having dimensions of $4 \times 3 \times 2$ meters. The studio background was of uniform blue color.

Human actions are represented by sequences of successive human body poses. In our experiments, the human body poses were binary body images, like the ones illustrated in Figure 7a. This is a widely adopted representation for human actions [46]. We used a color-based image segmentation technique, in

TABLE II

CLASSIFICATION RATES (%) IN UCR DATA SETS

Data set	Classes	Length	Train	Test	*K*NN	NB	C4_5	MLP	RandForces	LMT	SVM	Alg.#1	Alg.#2	Alg.#3
Adiac	37	176	390	391	59.34	56.78	53.2	**74.94**	57.8	72.12	43.99	68.54	68.54	70.59
50Words	50	270	450	455	64.4	56.26	41.76	**66.37**	55.16	56.92	64.62	64.18	64.4	65.49
CBF	3	128	30	900	85	89.67	67.33	85.33	83.56	77	87.67	98.33	99.67	**99.78**
ECG200	2	96	100	100	89	77	72	84	81	82	81	**98.33**	**98.33**	**98.33**
FaceAll	14	131	560	1690	68.64	69.17	55.03	82.43	60.95	75.74	71.83	77.75	83.49	**83.91**
FaceFour	4	350	24	88	87.5	84.09	71.59	87.5	78.41	77.27	**88.64**	76.14	80	80
fish	7	463	175	175	78.29	66.86	60	84	79.43	81.71	85.14	85.71	89.71	**90.86**
Gun Point	2	150	50	150	92	78.67	77.33	93.33	89.33	79.33	80	89.33	94.67	**95.33**
Lighting2	2	637	60	61	**80.33**	67.21	62.3	73.77	78.69	63.93	72.13	78.69	78.69	78.69
Lighting7	7	319	70	73	63.01	64.38	54.79	64.38	56.16	64.38	**71.23**	68.49	69.71	**71.23**
OSULeaf	6	427	200	242	**54.55**	37.19	36.78	44.63	41.74	49.17	43.8	48.38	51.56	52.07
SwedishLeaf	15	128	500	625	79.68	85.44	65.6	**86.56**	77.76	82.56	84.16	81.92	81.92	83.68
synthetic control	6	60	300	300	88	96	81	91.33	86	92	92.33	94.67	95.67	**96.33**
Trace	4	275	100	100	82	80	74	77	81	76	73	**100**	**100**	**100**
Two Patterns	4	128	1000	4000	90.60	45.68	65.13	89.65	72.5	83.23	82.2	92.83	93.5	**94.35**
wafer	2	152	1000	6147	99.4	70.83	98.2	96.28	99.32	98.09	95.96	98.8	99.38	**99.46**
yoga	2	246	300	3000	**83.3**	54.23	69.9	74.5	77.87	71.87	63.07	67.63	68.6	71.57
Beef	5	470	30	30	60	50	56.67	73.33	50	**80**	66.67	56.67	60	66.67
Coffee	2	286	28	28	75	67.86	57.14	96.43	75	**100**	96.43	**100**	**100**	**100**
OliveOil	4	570	30	30	76.67	76.67	73.33	**86.67**	**86.67**	83.33	**86.67**	73.33	76.67	80

Fig. 7. *a) Human body poses consisting a walking step captured by a side viewing angle and b) human body pose captured by multiple viewing angles.*

order to segment the body images from the blue background color. These videos were manually temporally segmented to produce smaller videos depicting one action instance each, e.g., one walk step. The binary body images were centered at the person's body center of mass and the maximum bounding box that encloses the person's body was used to produce binary videos depicting the human body poses. Human body poses captured from all viewing angles corresponding to the same time instance were combined, in order to produce binary images depicting the same human body pose from different viewing angles. An example of such images is illustrated in Figure 7b.

The resulting images were rescaled to 5×40 pixels and vectorized in a column-wise manner to produce 200-dimensional vectors. Thus, we assumed that an action is a sequence $\mathbf{s}_n$ consisting of instances $\mathbf{s}_{nm} \in$

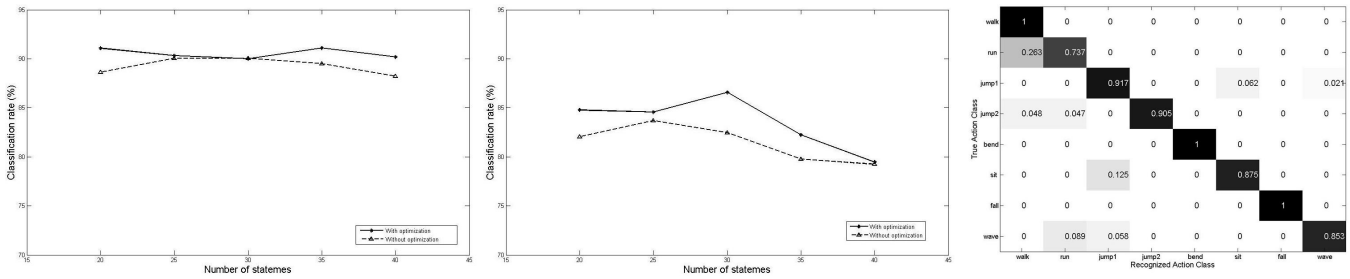

Fig. 8. *Classification rates in human action recognition using a) one subclass and b) two subclasses per class. c) Confusion matrix containing classification rates (%) in action recognition on the i3DPost database.*

$\mathbb{R}^{200}$. Since the duration of different actions varies, the number of instances forming each action sequence $\mathbf{s}_n$ is not constant.

All three optimization procedures have been used in the experiments conducted on the i3DPost database. In order to assess the ability of the proposed method to correctly classify data, where it has not been trained on, we performed the leave-one-person-out cross-validation procedure. That is, we used the action videos depicting seven persons to train the algorithm and the action videos depicting the eighth person for testing. This was applied eight times, one for each test person, to complete one experiment. The mean classification rate was computed to indicate the performance of the algorithm in one experiment. Multiple experiments have been performed, for different numbers of statemes and subclasses per class. In order to evaluate the contribution of the optimization procedures on the classification accuracy of the proposed method, a second set of experiments has been performed by applying no optimization at all. Figure 8 illustrates the classification rates observed for different numbers of statemes for both sets of experiments using one and two subclasses per class.

As can be seen in Figure 8, high classification rates have been obtained. Furthermore, it can be seen that the method including all the optimization steps provides higher sequence classification rates for all the presented experiments. The optimal parameters were found to be 20 statemes and one subclass per class, providing a classification rate equal to 91.08%. The confusion matrix corresponding to these parameters is illustrated in Figure 8c. In this Table, a row refers to the actual action class label, while a column refers

TABLE III

COMPARISON RESULTS IN THE I3DPOST ACTION RECOGNITION DATABASE.

	Holte et.al. [47]	Gkalelis et.al. [48]	**Proposed Method**
6 actions	89.58%	-	**90.88%**
5 actions	-	90%	**92.41%**

to the obtained classification result provided by the algorithm.

In order to compare our method with action recognition methods proposed in the literature using the i3DPost multi-view action recognition database for evaluation, we conducted experiments using fewer action classes. That is, we have performed the leave-one-person-out cross-validation procedure by used 6 actions ('walk', 'run', 'jump1', 'jump2', 'bend' and 'wave') and 5 actions ('walk', 'run', 'jump1', 'jump2' and 'bend') in order to compare the performance of our method with the performance of the methods presented in [47] and [48], respectively. Table III illustrates the comparison results. As can be seen, the proposed method outperforms both these action recognition methods.

IV. DISCUSSION

Stateme-based sequence representation provides several advantages, such as sequence duration invariance and application-independent sequence representation. However, there are two issues that should be properly addressed for efficient stateme based sequence representation.

The first issue is related to the optimal number of statemes. In the experiments presented in this paper, the optimal number of statemes has been determined by performing the N-fold cross-validation procedure. This is a widely adopted automatic parameter tuning procedure, which has the advantage that the optimal parameter values are determined by the training data at hand. However, it is a time consuming procedure, since the algorithm should be trained multiple times in a trial-and-error sense. The use of clustering techniques that are able to determine the optimal number of clusters [49], or iterative schemes determining the optimal number of clusters based on the training, like the one proposed in [50], could be a good alternative solution. However, such methods operate in an unsupervised manner and,

thus, the obtained number of statemes may not be the optimal for sequence classes discrimination. The determination of the optimal statemes number remains an open issue and could be an interesting research direction.

The second issue is related to the stateme definition. In our experiments we assumed that statemes are representative instances of sequential data. For example, in handwritten character recognition, we assumed that each character is a sequence of 2D contour pixel coordinates $[x, y]^T$. Thus, statemes were determined to be representative 2D points. Another choice could be to perform handwritten character recognition by using each character as one sample. This is the extreme case of using sequences formed by one instance only. In this case, statemes would be N-dimensional vectors, where $N = HW$ and H, W are the characters height and width, respectively. Another important issue is the choice of the similarity measure. As has been shown in the previously presented experiments, sequence representation based on the normalized mean fuzzy distance from the statemes is a good choice for real-valued instances. However, there are sequence classification problems, like protein sequence classification in genomic research, where such similarity measures may not be appropriate. In such cases, the similarity measure employed for statemes determination and sequence representation should contain sufficient information concerning the sequence properties.

V. Conclusion

In this paper, we proposed a multidimensional sequence classification method based on a novel sequence representation. This representation is based on the similarity of its instances from optimal instance prototypes, called statemes. Additionally, we proposed a modified Clustering Discriminant Analysis algorithm, determining both the optimal data projection matrix and representative class vectors, in terms of sequence classes discrimination, for discriminant sequence representation. The motivation of the proposed modified CDA algorithm is based on the observation that optimized subclass representative vectors, that may differ from the subclass mean vectors, could increase class discrimination. The aforementioned procedures provide a powerful sequence classification method dealing with challenges appearing in a wide variety of

sequence classification problems. The method has been tested on sequence classification databases coming from a wide range of applications, including handwritten character recognition, time series classification and human action recognition, giving very satisfactory results.

ACKNOWLEDGMENT

This work has been funded by the Collaborative European Project MOBISERV FP7-248434 (http://www.mobiserv.eu), An Integrated Intelligent Home Environment for the Provision of Health, Nutrition and Mobility Services to the Elderly.

REFERENCES

[1] A. Kampouraki, G. Manis and C. Nikou, "Heartbeat time series classification with suport vector machines", IEEE Transactions on Information Technology in Biomedicine, vol. 13, no. 4, pp. 512–518, 2009.

[2] L. Wei and E. Keogh, "Semi-supervised time series classification", ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 748–753, 2006.

[3] M. Vlachos, Z. Vagena, P.S. Yu, and V. Athitsos, "Rotation invariant indexing of shapes and line drawings", ACM International Conference on Information and Knowledge Management, pp. 131–138, 2005.

[4] A. Perina, M. Cristani, U. Castellani and V. Murino, "A new generative feature set based on entropy distance for discriminative classification", Image Analysis and Processing (ICIAP), pp. 199–208, 2009.

[5] X. Ji and H. Liu, "Advances in view-invariant human motion analysis: a review", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 40, no. 1, pp. 13–24, 2010.

[6] R.O. Duda, P.E. Hart and D.G. Stork, "Pattern Classification, 2nd ed", Wiley-Interscience, 2000.

[7] Z. Xing, J. Pei and E. Keogh, "A brief survey on sequence classification", , vol. 12, no. 1, pp. 40–48, 2010.

[8] W. Liao, "Clustering of time series data–a survey", Pattern Recognition, vol. 38, no. 11, pp. 1857–1874, 2005.

[9] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration", Data Mining and Knowledge Discovery, vol. 7, no. 4, pp. 349–371, 2003.

[10] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series", Workshop on Knowledge Discovery in Databases, pp. 359–370, 1994.

[11] E.J. Keogh and M.J. Pazzani, "Scaling up dynamic time warping for datamining applications", ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 285–289, 2000.

[12] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, E. and Keogh, "Indexing multidimensional time-series", International Journal on Very Large Data Bases, vol. 15, no. 1, pp. 1–20, 2006.

[13] M. Vlachos, G. Kollios and D. Gunopulos, "Elastic translation invariant matching of trajectories", Machine Learning, vol. 58 , no. 2, pp. 301–334, 2005.

[14] H.C. Yau and M.T. Manry., "Iterative improvement of a nearest neighbor classifier", Neural Networks, vol. 4, no. 4, pp. 517–524, 1991.

[15] D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval", European Conference on Machine Learning (ECML), pp. 4–15, 1998.

[16] B.Y.M. Cheng, J.G. Carbonell and J. Klein-Seetharaman, "Protein classification based on text document classification techniques", Proteins: Structure, Function, and Bioinformatics, vol. 58, no. 4, pp. 955-970, 2005.

[17] S.B. Kim, K.S. Han, H.C. Rim and S.H. Myaeng, "Some effective techniques for naive bayes text classification", IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 11, pp. 1457–1466, 2006.

[18] S.R. Eddy, "Hidden markov models", Current opinion in structural biology, vol. 6, no. 3, pp. 361–365, 1996.

[19] O. Yakhnenko, A. Silvescu and V. Honavar, "Discriminatively trained markov model for sequence classification", International Conference on Data Mining, pp. 498–505, 2005.

[20] J. Lafferty, A. McCallum and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", Machine Learning International Workshop, pp. 282–289, 2001.

[21] L. Ye and E. Keogh, "Time series shapelets: a new primitive for data mining", ACM SIGKDD International Conference on Knowledge Discovery and Data Minin, pp. 947–956, 2009.

[22] C.C. Aggarwal, "On effective classification of strings with wavelets", ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 163–172, 2002.

[23] A. Iosifidis, N. Nikolaidis and I. Pitas, "Movement recognition exploiting multi-view information", IEEE International Workshop on Multimedia Signal Processing (MMSP), pp. 427–431, 2010.

[24] G. Csurka, C.R. Dance, L. Fan, J. Willamowski and C. Bray, "Visual categorization with bags of keypoints", European Conference on Computer Vision (ECCV), pp. 1-22, 2004.

[25] A. Ramanan and M. Niranjan, "A Review of Codebook Models in Patch-Based Visual Object Recognition", Journal of Signal Processing Systems, pp. 1–20, 2011.

[26] F. Perronnin, "Universal and adapted vocabularies for generic visual categorization", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 7, pp. 1243–1261, 2008.

[27] F. Perronin, C. Dance, G. Csurka and M. Bressan, "Adapted vocabularies for generic visual categorization", European Conference on Computer Vision (ECCV), pp. 464–475, 2006.

[28] J. Winn, A. Criminisi and T. Minka, "Object categorization by learned universal visual dictionary", International Conference on Computer Vision (ICCV), pp. 1800–1807, 2005.

[29] W.H. Hsu and S.F. Chang, "Visual cue cluster construction via information bottleneck principle and kernel density estimation", pp. 82–91, 2005.

[30] F. Moosmann, F. Triggs and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests", Advances in Neural

Information Processing Systems (NIPS), pp. 985–992, 2006.

[31] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 7, pp. 1294–1309, 2009.

[32] X. Chen and T. Huang, "Facial expression recognition: a clustering-based approach", Pattern Recognition Letters, vol. 24, no. 10, pp. 1295–1302, 2003.

[33] A.R. Webb, "Statistical Pattern Recognition, 2nd ed", Wiley, 2002.

[34] J.C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", New York: Plenum, 1981.

[35] T. Kohonen, "The self-organizing map", Proceedings of the IEEE, vol. 78, no. 9, pp. 1464–1480, 2002.

[36] A. Ng, M. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm", Advances in Neural Information Processing Systems Conference, pp. 849–856, 2001.

[37] U. Luxburg, "A tutorial on spectral clustering", Statistics and Computing, vol. 17, no. 4, pp. 395–416, 2007.

[38] C. Blake and C.J. Merz, "{UCI} Repository of machine learning databases", 1998.

[39] B. Williams, M. Toussaint and A. Storkey, "Extracting motion primitives from natural handwriting data", International Conference on Artificial Neural Networks (ICANN), pp. 634–643, 2006.

[40] T.S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers", Advances in Neural Information Processing Systems, pp. 487–493, 1993.

[41] K. Tsuda, M. Kawanabe, G. Ratsch, S. Sonnenburg and K.R. Muller, "A new discriminative kernel from probabilistic models", Neural Computation, vol. 14, no. 10, pp. 2397–2414, 2002.

[42] E. Keogh, X. Xi, L. Wei and C.A. Ratanamahatana, "The UCR time series classification/clustering homepage", 2008.

[43] P.F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 306–318, 2008.

[44] V.L. Berardi and G.P. Zhang, "An empirical investigation of bias and variance in time series forecasting: modeling considerations and error evaluation", IEEE Transactions on Neural Networks, vol. 14, no. 3, pp. 668–679, 2003.

[45] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis and I. Pitas, "The i3DPost multi-view and 3D human action/interaction database", Conference on Visual Media Production, pp. 1511-1521, 2009.

[46] P. Turaga, R. Chellappa, V.S. Subrahmanian and O. Udrea, "Machine recognition of human activities: A survey", IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 11, pp. 1473–1488, 2008.

[47] M. Holte, T. Moeslund, N. Nikolaidis and I. Pitas, "3D Human Action Recognition for Multi-View Camera Systems", Joint Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), pp. 342–349, 2011.

[48] N. Gkalelis, N. Nikolaidis and I. Pitas, "View indepedent human movement recognition from multi-view video exploiting a circular invariant posture representation", IEEE International Conference on Multimedia and Expo (ICME), pp. 394–397, 2009.

[49] A. Likas, N. Vlassis and J.J. Verbeek, "The global k-means clustering algorithm", Pattern Recognition, vol. 36, no. 1, pp. 451–461, 2003.

[50] J.F. Brendan and D. Delbert, ”Clustering by Passing Messages Between Data Points”, Science, vol. 315, pp. 972–976, 2007.