Temporal Archetypal Analysis for Action Segmentation

Eftychia Fotiadou[1], Yiannis Panagakis[1,2], Maja Pantic[1,3]
[1] Department of Computing, Imperial College London, UK
[2] Department of Computer Science, Middlesex University, London, UK
[3] EEMCS, University of Twente, The Netherlands

Abstract— **Unsupervised learning of invariant representations that efficiently describe high-dimensional time series has several applications in dynamic visual data analysis. Clearly, the problem becomes more challenging when dealing with multiple time series arising from different modalities. A prominent example of this multimodal setting is the human motion which can be represented by multimodal time series of pixel intensities, depth maps, and motion capture data. Here, we study, for the first time, the problem of unsupervised learning of temporally and modality invariant informative representations, referred to as archetypes, from multiple time series originating from different modalities. To this end a novel method, coined as temporal archetypal analysis, is proposed. The performance of the proposed method is assessed by conducting experiments in unsupervised action segmentation. Experimental results on three different real world datasets using single modal and multimodal visual representations indicate the robustness and effectiveness of the proposed methods, outperforming compared state-of-the-art methods by a large, in most of the cases, margin.**

I. Introduction

Learning a *temporally invariant* subset of data points, referred to as *representatives* or *archetypes*, which can efficiently describe high-dimensional time series is an important (visual) data analysis problem. Temporally invariant archetypes are essentially the most informative slow-varying, data points of the time series and thus they can be used for summarization, representation, clustering, and segmentation of high-dimensional time series, such as videos. Representing time-varying data with a small number of archetypes has several advantages over working with long high-dimensional time series. First, archetypes facilitate the removal of outliers since they are not true representatives of the data. Moreover, the performance, the memory requirement, and the computational cost of clustering and segmentation algorithms is improved. The problem of learning temporally invariant archetypes becomes rather challenging when dealing with multiple time series arising from different modalities. For instance, human motion can be represented by multimodal time series of pixel intensities, depth maps, and motion capture data [1]. Similarly, a particular human behaviour can be identified by certain vocal, gestural, and facial features extracted from both the audio and visual modalities [2]. In this multimodal setting, the task is to find slow varying or temporally invariant prototypical data points efficiently describing the multiple time series with the additional property of being invariant across different modalities.

Even though learning of archetypes from high-dimensional (multimodal) time series has not be studied before, the problem of finding representative data points in static, non-time-varying, data is well-investigated and several methods have been proposed [3],[4],[5],[6],[7],[8],[9],[10],[11]. These methods can be organised into two categories based on their assumption on the underlying data generation model. The first category contains methods that assume that the data lie in a single low-dimensional (i.e., low-rank) subspace and find a subset of columns of the data matrix that corresponds to the best conditioned sub-matrix [3],[4]. The second category consists of methods assuming the data are drawn from multiple independent subspaces (or union of subspaces) and find representative data points by trying to approximate the data as non-negative [6],[7], sparse [5], or convex combination [8],[9],[10],[11] of a few other points in the union of subspaces. This property is known as self-expressiveness (e.g., [5]), which has been exploited in subspace clustering methods [12], [13], [14], [15]. Moreover these representatives data points approximate the convex-hull of the data [5],[8],[9],[10],[11].

However, none of the aforementioned methods account for the sequential relationships between successive data points in the high-dimensional time series and thus they are not able to derive temporally invariant data representations. To account for temporally invariant features or components, subspace learning methods can be applied. More specifically, by assuming that data come from a single low-rank model, the slow feature analysis (SFA) [16] and its variants extract slowly varying components from high-dimensional time series. In case of data drawn from a union of subspaces, temporal subspace segmentation methods can be employed [17],[18],[19]. However, these algorithms are designed to extract low-dimensional features rather than finding representatives or archetypical data points.

Here, distinct from the previous methods, the *temporal archetypal analysis* is proposed, enabling the discovery of slow-varying and modality invariant data representatives from multiple high-dimensional time series. In particular, we seek to express each data point in each time series as a convex combination of slowly-varying archetypes with the combination coefficients being shared among the different modalities. Moreover, the archetypes of each time series are also restricted to be convex combinations of the data. To find such invariant archetypes, a novel constrained optimization problem is solved by employing an iterative algorithm with

978-1-5090-4023-0/17/$31.00 ©2017 IEEE

guaranteed convergence.

The performance of the proposed method is assessed by conducting experiments in unsupervised action segmentation by employing three different datasets. In particular, the temporal archetypal analysis is tested on both single- and multimodal data. Experimental results indicate the effectiveness of the proposed approach on this application, outperforming state-of-the-art compared methods.

Notations. Throughout the paper, matrices (vectors) are denoted by uppercase (lowercase) boldface letters e.g., $\mathbf{X}$, ($\mathbf{x}$). The i-th column of $\mathbf{X}$ is denoted as $\mathbf{x}_i$, while the vector of ones of compatible dimensions as $\mathbf{1}$. Matrix elements are denoted with indexed lowercase letters, e.g., $(X)_{ij} = x_{ij}$. The set of real numbers is denoted by $\mathbb{R}$. A set of N real matrices of varying dimensions is denoted by $\{\mathbf{X}^{(n)} \in \mathbb{R}^{I_n \times J_n}\}_{n=1}^{N}$. The Frobenius norm is denoted by $\|\mathbf{X}\|_F = \sqrt{\sum_i \sum_j x_{ij}^2} = \sqrt{\text{tr}(\mathbf{X}^T\mathbf{X})}$ where $\text{tr}(\cdot)$ is the trace of a square matrix. The Euclidean norm of a vector is denoted by $\|\mathbf{x}\|_2$.

II. Methodology

In this section we present the proposed learning framework in detail. Moreover, we discuss how it can be used for segmentation of temporal sequences.

A. Temporal Archetypal Analysis

Let us assume that the visual stream is captured by different attribute types (modalities or views). Such multimodal cases arise frequently in real world computer vision applications. As an example, multiple cues of human action and behaviour can be captured, by employing multimodal time series (e.g., pixel intensities, depth maps, motion capture data, or data captured from different views). Combining information from multiple sources can lead to richer data representations and improved performance of various machine learning tasks.

Let us consider a multimodal dataset, denoted by $\{\mathbf{X}^{(n)} \in \mathbb{R}^{d_n \times T}\}_{n=1}^{N}$, where each matrix corresponds to a different modality. We assume that in each modality, the data can be described using a number of representatives. We can, thus, use an expression of the form $\mathbf{X}^{(n)} \approx \mathbf{A}^{(n)}\mathbf{S}$, where $\{\mathbf{A}^{(n)} \in \mathbb{R}^{d_n \times k}\}_{n=1}^{N}$ contain k representatives associated to the n-th modality as column vectors, while $\mathbf{S}$ contains the reconstruction coefficients. Another aim of our approach is to seek for representatives that resemble the original data and, therefore, provide a more intuitive and informative description. To this end, we impose the additional constraint that the representatives must be convex combinations of the data points. Additionally, each data point is expressed as a convex combination of the representatives. These properties, which also introduce a symmetry between the representatives and the data points, were also employed in the archetypal analysis method [8], where the extracted representatives are referred to as *archetypes*.

Taking the aforementioned constraints into account, the n-th modality data matrix is expressed as: $\mathbf{X}^{(n)} \approx \mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S}$, where $\mathbf{C}^{(n)}$ and $\mathbf{S}$ are column stochastic matrices (i.e., the elements of each column are nonnegative and sum to one). It should be noted that matrix $\mathbf{S}$ is shared among the different modalities, implying that a common low-dimensional and modality invariant representation of the data is learned.

The above discussion leads us to the formulation of a learning framework, expressed by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{C}^{(n)},\mathbf{S}} & \sum_n \|\mathbf{X}^{(n)} - \mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S}\|_F^2 \\ \text{s.t.} \quad & \mathbf{C}^{(n)} \geq 0,\ \mathbf{S} \geq 0,\ \mathbf{1}^T\mathbf{C}^{(n)} = \mathbf{1}^T,\ \mathbf{1}^T\mathbf{S} = \mathbf{1}^T, \end{aligned} \tag{1}$$

where $n = 1, ..., N$, and the constraints ensure that the representatives are convex combinations of the observations, as well as that the observations are approximated by convex combinations of the representatives.

As already mentioned, our objective is to extract temporally invariant representations, or at least representations which vary slowly in time. In order to achieve this a temporal regularization term $\tau(\mathbf{S})$ operating upon the representation matrix $\mathbf{S}$ is incorporated to the objective function, which takes the form:

$$\min_{\mathbf{C}^{(n)},\mathbf{S}} \sum_n \|\mathbf{X}^{(n)} - \mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S}\|_F^2 + \tau(\mathbf{S}) \tag{2}$$

The temporal term can be expressed as an approximation of the first order derivative of the representation matrix, calculated by:

$$\tau(\mathbf{S}) = \|\mathbf{SP}\|_F^2 = \text{tr}(\mathbf{SLS}^T), \tag{3}$$

where $\mathbf{P} \in \mathbb{R}^{T \times (T-1)}$ with $P_{ii} = -1, P_{i+1,i} = 1$, a matrix encoding forward differences between columns of $\mathbf{S}$ and $\mathbf{L} = \mathbf{PP}^T$. This choice is motivated by SFA [16], a method for identifying invariant or slowly varying features in dynamic data. By combining (2) and (3), the final optimization problem takes the form:

$$\begin{aligned} \min_{\mathbf{C}^{(n)},\mathbf{S}} & \sum_n \|\mathbf{X}^{(n)} - \mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S}\|_F^2 + \lambda tr(\mathbf{SLS}^T) \\ \text{s.t.} \quad & \mathbf{C}^{(n)} \geq 0,\ \mathbf{S} \geq 0,\ \mathbf{1}^T\mathbf{C}^{(n)} = \mathbf{1}^T,\ \mathbf{1}^T\mathbf{S} = \mathbf{1}^T, \end{aligned} \tag{4}$$

where $\lambda > 0$ is a trade-off parameter between the approximation and the regularization terms.

For the solution of the minimization problem (4), the algorithmic framework proposed in [20] is employed. This framework is based on the multiplicative update rules method for solving the Nonnegative Matrix Factorization (NMF) problem, originally introduced in [21]. Although the multiplicative updates of [21] guarantee that the objective function is non-increasing, it has been shown that they do not guarantee convergence to a stationary point [22]. Furthermore, their convergence can be very slow. The algorithmic framework presented in [20] tackles the aforementioned shortcomings of the multiplicative updates, providing a fast implementation for the NMF solution and convergence to a stationary point. This is achieved by introducing in each iteration an additional loop of multiplicative updates for each variable, where the same variable is updated multiple times before the update of the other.

To solve the minimization problem in (4), the associated Lagrangian function is expressed as:

$$\begin{aligned}\mathcal{L}(\mathcal{V}) = &\sum_n \mathrm{tr}(\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}) - 2\sum_n \mathrm{tr}(\mathbf{S}^T\mathbf{C}^{(n)^T}\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}) \\ &+ \sum_n \mathrm{tr}(\mathbf{S}^T\mathbf{C}^{(n)^T}\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S}) + \lambda \mathrm{tr}(\mathbf{S}\mathbf{L}\mathbf{S}^T) \\ &+ \sum_n \mathrm{tr}(\mathbf{\Phi}^{(n)}\mathbf{C}^{(n)^T}) + \mathrm{tr}(\mathbf{\Psi}\mathbf{S}^T),\end{aligned} \tag{5}$$

where $\mathcal{V} = \{\{\mathbf{C}^{(n)}\}_{n=1}^{N}, \mathbf{S}, \{\mathbf{\Phi}^{(n)}\}_{n=1}^{N}, \mathbf{\Psi}\}$, and $\mathbf{\Phi}^{(n)}$ and $\mathbf{\Psi}$ are Lagrangian multipliers corresponding to the nonnegativity constraints. By differentiating with respect to $\mathbf{C}^{(n)}$ we have:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{C}^{(n)}} = 2\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S}\mathbf{S}^T - 2\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{S}^T + \mathbf{\Phi}^{(n)} \tag{6}$$

Similarly, the partial derivative of the Lagrangian with respect to $\mathbf{S}$ is calculated by:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{S}} = &\,2\sum_n \mathbf{C}^{(n)^T}\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S} \\ &- 2\sum_n \mathbf{C}^{(n)^T}\mathbf{X}^{(n)^T}\mathbf{X}^{(n)} + 2\lambda\mathbf{S}\mathbf{L} + \mathbf{\Psi}\end{aligned} \tag{7}$$

From the Karush-Kuhn-Tucker (KKT) conditions [23], it holds that $\phi_{ij}^{(n)}c_{ij}^{(n)} = 0$, $\psi_{ij}s_{ij} = 0$, and also:

$$\begin{aligned}&(\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S}\mathbf{S}^T)_{ij}c_{ij}^{(n)} - (\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{S}^T)_{ij}c_{ij}^{(n)} = 0 \\ &\sum_n (\mathbf{C}^{(n)^T}\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S})_{ij}s_{ij} \\ &-\sum_n (\mathbf{C}^{(n)^T}\mathbf{X}^{(n)^T}\mathbf{X}^{(n)})_{ij}s_{ij} + \lambda(\mathbf{S}\mathbf{L})_{ij}s_{ij} = 0.\end{aligned} \tag{8}$$

The last two equations lead to the following multiplicative update rules:

$$\begin{aligned}c_{ij}^{(n)}[t] &\leftarrow c_{ij}^{(n)}[t-1]\frac{(\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{S}^T)_{ij}}{(\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S}\mathbf{S}^T)_{ij}} \\ s_{ij}[t] &\leftarrow s_{ij}[t-1]\frac{(\sum_n \mathbf{C}^{(n)^T}\mathbf{X}^{(n)^T}\mathbf{X}^{(n)} - \lambda\mathbf{S}\mathbf{L}^-)_{ij}}{(\sum_n \mathbf{C}^{(n)^T}\mathbf{X}^{(n)^T}\mathbf{X}^{(n)}\mathbf{C}^{(n)}\mathbf{S} + \lambda\mathbf{S}\mathbf{L}^+)_{ij}},\end{aligned} \tag{9}$$

where matrices $\mathbf{L}^+$ and $\mathbf{L}^-$ are defined as $\max(\mathbf{L}, 0)$ and $\min(\mathbf{L}, 0)$ respectively and operations are applied element-wise. The procedure followed for solving the optimization problem in (4) is summarised in Algorithm 1. In the plots of Fig. 1, the relative reconstruction error (i.e., the ratio of the reconstruction error $e(t)$ at time t to the initial reconstruction error $e(0)$) as a function of iterations is depicted. The decreasing behaviour of the relative reconstruction error indicates the good convergence properties of the proposed algorithm.

Algorithm 1 Temporal Archetypal Analysis

```
Require: Data: ${\mathbf{X}^{(n)} \in \mathbb{R}^{d_n \times T}}_{n=1}^{N}$. Parameters: $\lambda, k$, $tol$ (tolerance)
Initialization: Initialize matrices ${\mathbf{C}^{(n)}}_{n=1}^{N}, \mathbf{S}$ with random values, set parameter values $\alpha$, $\delta$, $\rho_C$, $\rho_S$ following [20].
for $t = 0, 1, 2, ...$ do
    for $n = 1, 2, ..., N$ do
        for $m = 1 : 1 + \alpha\rho_C$ do
            Update $\mathbf{C}^{(n)}[t, m]$ from (9)
            if $\|\mathbf{C}^{(n)}[t,m] - \mathbf{C}^{(n)}[t,m-1]\|_F \leq \delta\|\mathbf{C}^{(n)}[t,1] - \mathbf{C}^{(n)}[t,0]\|_F$ then
                break ;
            end if
        end for
        $\mathbf{C}^{(n)}[t+1] \leftarrow \mathbf{C}^{(n)}[t,m]$
        Normalize columns of $\mathbf{C}^{(n)}[t+1]$ to unit sum
    end for
    for $l = 1 : 1 + \alpha\rho_S$ do
        Update $\mathbf{S}[t, l]$ from (9)
        if $\|\mathbf{S}[t,l] - \mathbf{S}[t,l-1]\|_F \leq \delta\|\mathbf{S}[t,1] - \mathbf{S}[t,0]\|_F$ then
            break ;
        end if
    end for
    $\mathbf{S}[t+1] \leftarrow \mathbf{S}[t,l]$
    Normalize columns of $\mathbf{S}[t+1]$ to unit sum
    $\mathrm{e}^{(n)}[t{+}1] = \|\mathbf{X}^{(n)} - \mathbf{X}^{(n)}\mathbf{C}^{(n)}[t+1]\mathbf{S}[t+1]\|_F / \|\mathbf{X}^{(n)}\|_F$
    $\mathrm{err}[t+1] = \max_n(e^{(n)}[t+1])$
    if $|\mathrm{err}[t+1] - \mathrm{err}[t]| \leq tol$ then
        break;
    end if
end for
```

B. Temporal Segmentation of High-Dimensional Time Series

Having found a common low-dimensional representation for the data from different modalities, it can be used in order to perform segmentation on the temporal sequence. In more detail, matrix $\mathbf{S}$ is used to construct a graph $\mathcal{G}$, with a corresponding symmetric affinity matrix $\mathbf{W}$ calculated by:

$$w_{ij} = \frac{\mathbf{z}_i^T\mathbf{z}_j}{\|\mathbf{z}_i\|_2\|\mathbf{z}_j\|_2}, \tag{10}$$

where $\mathbf{z}_i$ denotes the i-th column of matrix $\mathbf{Z} = \mathbf{S}^T\mathbf{S}$. The segmentation result is finally obtained by applying the Normalized Cuts (Ncut) algorithm [24] on the aforementioned graph.

III. Experimental Evaluation

In order to evaluate the performance of the proposed method we consider a temporal action segmentation scenario. Given a video depicting a subject performing multiple actions sequentially, the aim is to divide it into disjoint segments, each of which containing the frames corresponding to a single action. The proposed framework is evaluated in both single modality and multimodal experiments involving two or three modalities.

The performance of our method is compared against that of various state-of-the-art methods:

- Subspace clustering methods: Sparse Subspace Clustering (SSC) [12], Low-Rank Representation (LRR) [13],

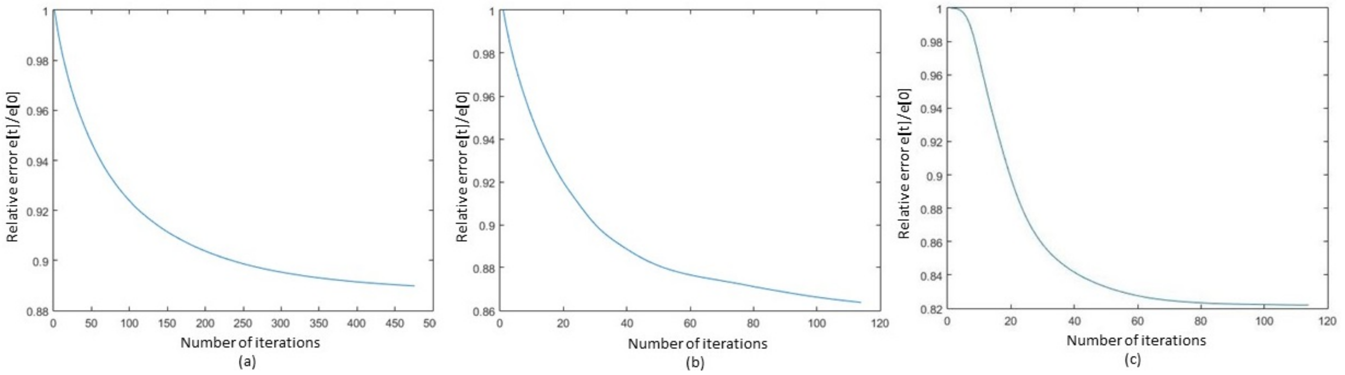

Fig. 1. Relative approximation error as a function of number of iterations for the proposed method applied on: (a) Weizmann, (b) Ballet, and (c) MAD (Depth data) datasets.

TABLE I

TEMPORAL ACTION SEGMENTATION RESULTS FOR THE WEIZMANN AND BALLET DATASETS (AVERAGE RATES)

Method	**Weizmann dataset**		**Ballet dataset**	
	ACC (%)	NMI (%)	ACC (%)	NMI(%)
SSC [12]	32.386	55.8	42.969	41.8
LRR [13]	30.942	54.2	43.344	41.8
LSSC [25]	30.069	54	44.5	43
LRSC [26]	30.717	53.4	44.406	41.7
AA [9]	31.809	54.9	43.938	42.4
PAA [11]	29.202	52.6	44.656	44.5
RAA [10]	31.11	53.2	42.984	42
SMRS [5]	31.832	55.2	44.547	43.5
OSC [18]	39.868	65.7	44.813	43.2
TSC [17]	39.429	64.8	73.625	74
GNMF [27]	42.755	64.3	42.953	40.6
Proposed method	**80.767**	**90**	**91.063**	**87.58**

Least Squares Subspace Clustering (LSSC) [25], Low Rank Subspace Clustering (LRSC) [26].

- Methods for extracting archetypes: Archetypal Analysis (AA) [9], Robust Archetypal Analysis (RAA) [10], Probabilistic Archetypal Analysis (PAA) [11], and Sparse Modeling Representative Selection (SMRS) [5].
- Temporal subspace learning methods: Temporal Subspace Clustering (TSC) [17], Ordered Subspace Clustering (OSC) [18], and Graph-regularized Nonnegative Matrix Factorization (GNMF) [27].

A. Datasets

For the purposes of our experiments, three different databases where used, namely the Weizmann database [28], the Ballet dataset [29],[30], and the Multimodal Action Database [1]. The two first were used for single modality segmentation, while the third one was used in multimodal experiments.

- **Weizmann database**: the Weizmann database contains video recordings of nine different subjects performing the following 10 actions: “run”, “walk”, “skip”, jumping-jack” (or “jack”), “jump-forward-on-two-legs” (or “jump”), “jump-in-place-on-two-legs”, “gallopsideways” (or “side”), “wave-two-hands (or “wave2”), “wave-one-hand (or “wave1”), or “bend”. In total, the database contains 90 videos (one video per subject per action), of resolution 180x144 pixels. Along with the video recordings, the binary masks of each frame obtained after background subtraction are also provided. Following [17], we extract a dictionary-based representation for every video frame. In more detail, a codebook of 100 codewords is calculated from the binary masks, by applying the k-means clustering algorithm. Subsequently, each frame in a video sequence is represented by a binary (indicator) vector with a unique unit entry in the index corresponding to the closest (in terms of Euclidean distance) codeword. Since each video recording of the Weizmann database depicts a single activity, in order to construct sequences suitable for segmentation, we concatenate multiple clips together. Following an experimental setting similar to that used in [17], we randomly select 5 sequences from each subject and concatenate them into a long one. The aforementioned procedure is repeated ten times, resulting in 10 long sequences, consisting of 45 actions each.
- **Ballet dataset**: the ballet dataset contains 44 videos collected from an instructional ballet DVD. Each frame in the dataset is annotated with one of the eight action labels: “left-to-right hand opening”, “right-to-left hand opening”, “standing hand opening”, “leg swinging”, “jumping”, “turning”, “hopping”, and “standing still”. The aforementioned actions are performed by three different subjects, two male and one female. It should be noted that the label descriptions in this dataset are rather general, meaning that strong variations exist within each action class. For example, videos characterized by the label ”standing hand opening” may contain significantly diverse movements of the legs. Intra-class variations are further induced by differences in execution style and speed, by the subjects’ clothing (the female subject is wearing a skirt), as well as by camera motion. For the above reasons, temporal action segmentation on the ballet data is much more challenging than on the Weizmann dataset.
In order to apply temporal segmentation on the ballet data, we follow a pre-processing procedure, similar to

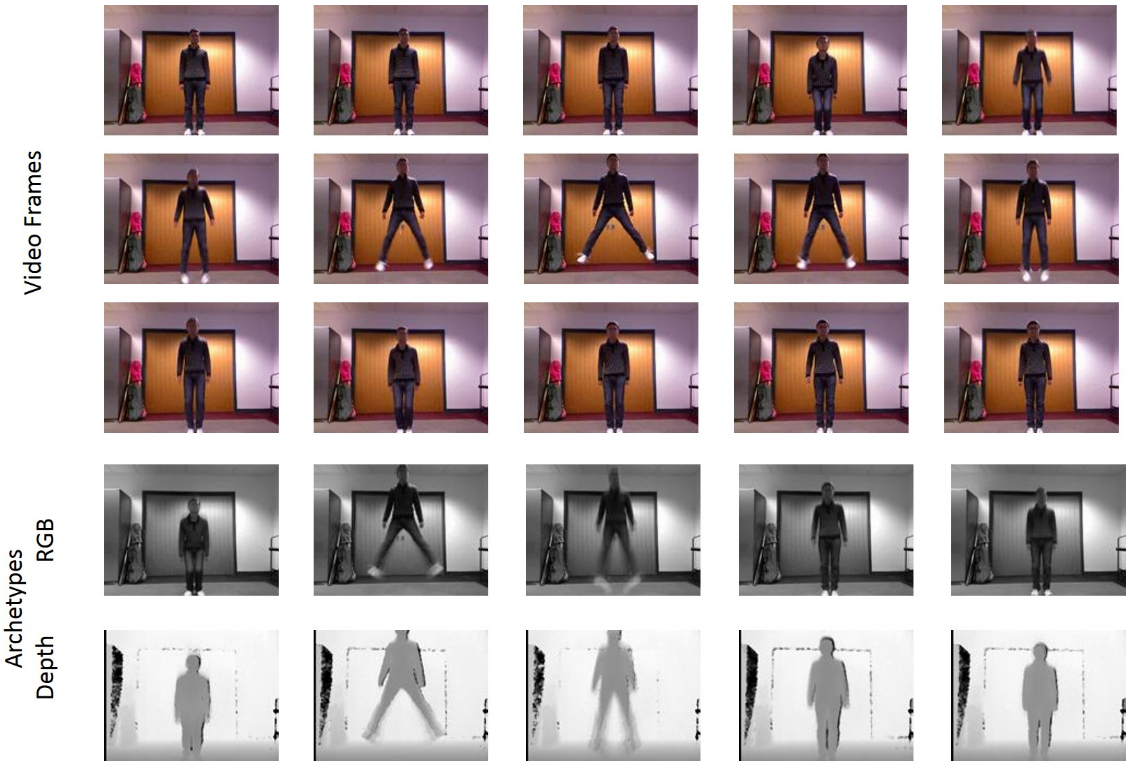

Fig. 2. Visual representation of archetypes jointly extracted from RGB and depth modalities. The top three rows depict 15 frames of action ”jumping jack” from the MAD dataset. In the fourth row archetypes extracted from the RGB modality are illustrated. In the last row archetypes extracted from the depth modality are shown. Clearly, the extracted archetypes of each modality capture the same elementary motions of the action.

that used in [19]. Initially, the videos are rescaled to 50×50 and each frame is vectorized. Subsequently, the frames from each video that belong to the same action class, are grouped in image sets, consisting of 20 frames each. Successive image sets of the same video are selected to have an overlap of 15 frames. As a result, each action class is associated with a number of image sets. By combining image sets from different classes, longer sequences are constructed. Specifically, we choose 4 image sets at random from each class and concatenate them, such that the final sequence is constructed as: $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_8]$. Using this procedure, 10 long sequences are obtained.

- **Multimodal Action Database**: this database contains human activities recorded using a Microsoft Kinect sensor in an indoor environment. The modalities include RGB video, 3D depth video, and skeleton data (3D coordinates of 20 joints for each frame). There are 20 subjects in total, performing 35 different actions sequentially, and each subject performs the same sequence of activities twice. For our experiments we randomly select 5 subjects and one trial from each of them.
The available modalities are combined in three different ways: RGB + depth data, RGB + skeleton data, and RGB + depth + skeleton data. RGB and depth video frames are rescaled by a factor equal to 0.15, and subsequently vectorized. Skeleton data are also vectorized. When used in experiments with the baseline methods, the different modalities are combined by feature concatenation, i.e. by concatenating the feature vectors from each modality into a single vector. In the case of RGB + skeleton or RGB + depth + skeleton experiments, and in order to obtain feature vectors of compatible dimensions and magnitudes, we first perform PCA on the RGB and depth data, retaining 100 components. Subsequently, both the RGB and skeleton data values are normalized in $[0, 1]$.

B. Experimental results

After a representation has been learned with one of the subspace methods, it is used to build an affinity matrix, upon which the NCut clustering algorithm is applied. For our method we follow the procedure presented in Section II-B, while for the methods used for comparisons the affinity matrices are constructed in the way reported in the original papers (where applicable).

The number of clusters in the NCut algorithm is set equal to the number of the individual action clips each sequence comprises of, i.e., 45 for the Weizmann, 8 for the ballet, and 35 for the MAD dataset. As evaluation measures, clustering accuracy (ACC) and normalized mutual information (NMI), averaged on all the sequences, are employed [31].

The proposed method involves two parameters, the optimal values of which have to be determined: the number of archetypes k and the trade-off parameter λ. To this end, grid search was performed and the values that yielded the highest

TABLE II

MULTIMODAL TEMPORAL ACTION SEGMENTATION RESULTS FOR THE MAD DATASET (AVERAGE RATES)

Method	**RGB + Depth**		**RGB + Skeleton**		**RGB + Depth + Skeleton**	
	ACC (%)	NMI (%)	ACC (%)	NMI (%)	ACC (%)	NMI (%)
SSC [12]	48.025	69.1	40.858	61	39.7	59.9
LRR [13]	38.355	58.2	38.988	59.2	36.878	54.5
LSSC [25]	38.136	58.3	38.509	59	36.11	54.9
LRSC [26]	37.82	56.4	39.922	59	37.33	53.7
AA [9]	35.972	53.1	35.522	53.3	34.258	50.6
PAA [11]	8.203	10.4	8.367	10.5	8.378	10.5
RAA [10]	37.345	57.9	38.843	58.8	35.353	53.8
SMRS [5]	37.614	56.9	38.288	58	35.629	54.2
OSC [18]	38.165	57.5	38.876	58.9	36.176	53.9
TSC [17]	55.347	74.7	71.331	85.64	**75.484**	**86.8**
GNMF [27]	28.212	45.6	27.71	41.9	25.889	37.5
Proposed method	**70.467**	**84.5**	**73.111**	**86.4**	74.742	86.5

clustering accuracy were selected. The optimal parameters of the baseline methods were also determined through grid search. In the case of GNMF, matrix $\mathbf{L}$ given in (3), was used as the Laplacian matrix of the regularization term, encoding, thus, the temporal information of the data.

In Table I, the results for the single modality temporal segmentation experiments are presented. The first four rows correspond to common subspace clustering method (SSC, LRR, LRSC, LSSC), while the subsequent four display the performance of algorithms which extract representatives from the data (i.e., AA, PAA, RAA, SMRS). The aforementioned methods do not make any assumptions about the temporal nature of the data, therefore, a method which takes temporal information into consideration, like the proposed one, is expected to perform better. As can be observed, the proposed method significantly outperforms these methods. However, our method exhibits a better performance compared to other subspace methods intended for temporal data, such as TSC, OSC, and GNMF.

Similar results were obtained for the multimodal experiments, as can be observed in Table II. In most cases the proposed method performed considerably better than the baseline methods, with the exception of RGB+depth+skeleton data combination, where the TSC method achieved a higher clustering accuracy. In order to examine the influence of combining multiple modalities on the segmentation result, we also performed experiments with the proposed method for each modality individually. As can be observed from Table III, the clustering accuracy is in most cases improved when two or three modalities are combined. When RGB are used along with depth data, a higher accuracy is obtained as compared to using only RGB data, while no significant improvement is observed when only depth data were used. On the other hand, the combination of RGB with skeleton data yields an improved performance in comparison to both single modality experiments (RGB only and Skeleton only). Finally, using all three modalities available further boosts the segmentation performance. A visual representation of the archetypes extracted using the RGB and depth modalities from a ”jumping jack” action sequence of the MAD dataset is illustrated in Fig. 2.

TABLE III

TEMPORAL ACTION SEGMENTATION RESULTS FOR THE PROPOSED METHOD. SINGLE MODALITY EXPERIMENTS WITH THE MAD DATASET

RGB		**Depth**		**Skeleton**	
ACC (%)	NMI (%)	ACC (%)	NMI (%)	ACC (%)	NMI (%)
66.349	83.2	70.851	84.5	69.287	84.6

IV. CONCLUSIONS

In this paper we have investigated the problem of finding temporal and modality invariant archetypes from multimodal high-dimensional time series. The proposed method combines two advantages, which render it efficient: first, it learns representatives that are convex combinations of actual data points, thus, characterizing the convex hull of the data. Furthermore, it considers the temporal relationships between successive observations, which leads to the extraction of temporally invariant representations. The proposed method was evaluated in a temporal action segmentation scenario applied on three different datasets, where it outperformed the compared state-of-the-art methods.

V. ACKNOWLEDGMENTS

This work is funded by the EPSRC project EP/N007743/1 (FACER2VM). The work of Yannis Panagakis and Maja Pantic is also partially supported by by the European Community Horizon 2020 [H2020/2014-2020] under grant agreement no. 645094 (SEWA).

REFERENCES

[1] D. Huang, Y. Wang, S. Yao, and F. De la Torre. Sequential max-margin event detectors. In *European Conference on Computer Vision (ECCV)*, 2014.

[2] Y. Panagakis, M. A. Nicolaou, S. Zafeiriou, and M. Pantic. Robust correlated and individual component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue in Multimodal Pose Estimation and Behaviour Analysis, (accepted)*, 2016.

[3] J. A. Tropp. Column Subset Selection, Matrix Factorization, and Eigenvalue Optimization. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’09, pages 978–986, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[4] C. Boutsidis, M. W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 968–977, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[5] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1600–1607, June 2012.

[6] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin. A convex model for nonnegative matrix factorization and dimensionality reduction on physical space. *IEEE Transactions on Image Processing*, 21(7):3239–3252, July 2012.

[7] C. H. Q. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):45–55, January 2010.

[8] A. Cutler and L. Breiman. Archetypal analysis. *Technimetrics*, 36(4):338–347, 1994.

[9] M. Mørup and L. K. Hansen. Archetypal analysis for machine learning and data mining. *Neurocomputing*, 80:54 – 63, 2012. Special Issue on Machine Learning for Signal Processing 2010.

[10] Y. Chen, J. Mairal, and Z. Harchaoui. Fast and Robust Archetypal Analysis for Representation Learning. In *CVPR 2014 - IEEE Conference on Computer Vision Pattern Recognition*, May 2014.

[11] S. Seth and M. J. Eugster. Probabilistic archetypal analysis. *Mach. Learn.*, 102(1):85–113, January 2016.

[12] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, pages 2790–2797, June 2009.

[13] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[14] Y. Panagakis, C. L. Kotropoulos, and G. R. Arce. Music genre classification via joint sparse low-rank representation of audio features. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 22(12):1905–1917, December 2014.

[15] Y. Panagakis and C. Kotropoulos. Elastic net subspace clustering applied to pop/rock music structure analysis. *Pattern Recognition Letters*, 38:46 – 53, 2014.

[16] L. Wiskott and T. J. Sejnowski. Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Comput.*, 14(4):715–770, April 2002.

[17] S. Li, K. Li, and Y. Fu. Temporal subspace clustering for human motion segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4453–4461, Dec 2015.

[18] S. Tierney, J. Gao, and Y. Guo. Subspace clustering for sequential data. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1026, June 2014.

[19] F. Wu, Y. Hu, J. Gao, Y. Sun, and B. Yin. Ordered subspace clustering with block-diagonal priors. *IEEE Transactions on Cybernetics*, PP(99):1–11, 2015.

[20] N. Gillis and F. Glineur. Accelerated Multiplicative Updates and Hierarchical Als Algorithms for Nonnegative Matrix Factorization. *Neural Comput.*, 24(4):1085–1105, April 2012.

[21] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001.

[22] C. J. Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 18(6):1589–1596, Nov 2007.

[23] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.

[25] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan. Robust and Efficient Subspace Segmentation via Least Squares Regression. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ECCV'12, pages 347–360, Berlin, Heidelberg, 2012. Springer-Verlag.

[26] R. Vidal and P. Favaro. Low rank subspace clustering (LRSC). *Pattern Recognition Letters*, 43:47 – 61, 2014. {ICPR2012} Awarded Papers.

[27] D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, Aug 2011.

[28] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as Space-Time Shapes. In *The Tenth IEEE International Conference on Computer Vision (ICCV'05)*, pages 1395–1402, 2005.

[29] Y. Wang and G. Mori. Human action recognition by semilatent topic models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1762–1774, 2009.

[30] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, pages 1–8, June 2008.

[31] W. Xu, X. Liu, and Y. Gong. Document clustering based on nonnegative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 267–273, New York, NY, USA, 2003. ACM.