Learning Low-Dimensional Temporal Representations

Bing Su [1] **Ying Wu** [2]

Abstract

Low-dimensional discriminative representations enhance machine learning methods in both performance and complexity, motivating supervised dimensionality reduction (DR) that transforms high-dimensional data to a discriminative subspace. Most DR methods require data to be i.i.d., however, in some domains, data naturally come in sequences, where the observations are temporally correlated. We propose a DR method called LT-LDA to learn low-dimensional temporal representations. We construct the separability among sequence classes by lifting the holistic temporal structures, which are established based on temporal alignments and may change in different subspaces. We jointly learn the subspace and the associated alignments by optimizing an objective which favors easily-separable temporal structures, and show that this objective is connected to the inference of alignments, thus allows an iterative solution. We provide both theoretical insight and empirical evaluation on real-world sequence datasets to show the interest of our method.

1. Introduction

Multivariate temporal sequences arise in a wide range of applications, where the pattern of interest is represented as a sequence of local feature vectors. The local features may be high-dimensional and contain noisy information. Thus it is desirable to reduce the dimension of the features in sequences by projecting them to a discriminative low-dimensional subspace, in which sequence classification would become faster and more accurate.

Various supervised *dimensionality reduction (DR)* methods have been developed for vector data under the i.i.d. assumption, but they cannot be applied to the features in sequences by omitting the temporal dependencies. DR for sequence data aims at learning a subspace by maximizing the separability among sequence classes, where the separability embodies in the differences on temporal structures. The temporal structures reflect the common evolutions of all sequences from the same class, and they depend on temporal alignments to establish correspondences among sequences with local temporal differences. The separability and objective are more difficult to formulate and manipulate by nature. For these reasons, DR for sequence data has received much less attention.

Existing methods such as *Linear Sequence Discriminant Analysis (LSDA)* (Su & Ding, 2013; Su et al., 2018) and *Max-Min inter-Sequence Distance Analysis (MMSDA)* (Su et al., 2017a) construct the separability based on generative models. For each class, they train a left-to-right *Hidden Markov Model (HMM)* (Rabiner, 1989)) from the original sequences. The mean of the features aligned to each hidden state is calculated, and the means of all ordered states form a mean sequence. The inter-class distance is measured as the *Dynamic Time Warping (DTW)* (Sakoe & Chiba, 1978) distance between the mean sequences. Such separability depends on the alignments between the sequences and the hidden states, which further rely on the similarities of the features. When projecting the features to a subspace, the local similarities among the transformed features may change, and hence the alignments may change accordingly. On the other hand, the projection is determined by maximizing the separability, where the separability should be constructed based on the alignments in the subspace. Therefore, learning the projection and inferring the alignments are entangled. To make it tractable, existing methods simply fix the alignments in the underlying subspace to those in the original space. However, the resulting separability cannot reflect the real confusion relationship between classes in the subspace. Also, HMM-based separability requires a large number of sequences for training and is poor in scalability.

In this paper, we propose a supervised DR method for sequence data called *Latent Temporal Linear Discriminant Analysis (LT-LDA)*. We learn an abstract template for each class to discover the temporal structures via employ-

[1]Science & Technology on Integrated Information System Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing, China [2]Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA. Correspondence to: Bing Su <subingats@gmail.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

ing the modified DTW barycenter (Petitjean et al., 2011; Su et al., 2016). We then construct the separability among sequence classes based on the alignments between the abstract templates and the training sequences. Although determining the alignments by learning the abstract templates and learning the subspace by maximizing the constructed separability still rely on each other, we show that their objectives are actually connected, which allows us to jointly learn the most discriminative subspace together with the associated latent alignments, resulting in sequences of low-dimensional discriminative temporal representations.

The main contributions are as follows. (1) Different from the HMM-based separability, our new construction of separability does not require lots of training data. It can be performed even when only one training sequence per class is available. (2) Different from previous methods where the subspace can only be learned through pre-fixing the alignments, we propose to learn the subspace and the latent alignments simultaneously and develop an efficient iterative solution. The learned subspace is thereby holistically optimal. (3) We establish a connection between our objective formulation and the abstract template learning, which ensures the convergence of our solution. We further provide theoretical insight on the subspace selection.

2. Related Work

Various supervised linear DR methods have been proposed for vector data, such as Linear Discriminant Analysis (LDA) (Fisher, 1936), Marginal Fisher Analysis (Yan et al., 2007), and Max-min Distance Analysis (Bian & Tao, 2011; Zhang & Yeung, 2010). LDA optimizing the Fisher criterion is perhaps the most widely used method for its simpleness, effectiveness and the well-established theory, and is getting consistent interest (De la Torre & Kanade, 2006; Ding & Li, 2007; Ye et al., 2007; Nikitidis et al., 2014) in machine learning. These methods cannot be applied to vectors in sequences, which violate the basic i.i.d. assumption. Our method performs DR for sequence data by lifting the inherent temporal dependencies.

In (Zhou & De la Torre, 2012; Trigeorgis et al., 2018), linear and non-linear transformations were learned for each sequence pair to perform multi-modal alignments. The transformations for different sequence pairs are different. In our method, the projection is for discriminating different classes and stays the same for all sequences from all classes. In (Shyr et al., 2010), a sufficient DR approach was proposed for sequence labeling by building sequence kernels. The labels are associated with the vectors in sequences rather than the whole sequences, and the task is to predict a class label for each vector in the sequences. In (Flamary et al., 2012), the features are transformed by unidimensional convolutions of all dimensions for sequence labeling. Our method focuses on linear projection and the task is to predict a label for each entire sequence. In (Lajugie et al., 2014), a Mahalanobis distance was learned given the ground-truth alignments of training samples for multivariate sequence alignment, while in our method the alignments of both the training sequences and the test sequences are unavailable.

LSDA (Su & Ding, 2013; Su et al., 2018) and MMSDA (Su et al., 2017a) targeted at the same problem as this paper, where the projection was learned by maximizing the separability defined on HMM-based temporal structures. The alignments of the sequences to the hidden states in the original space and the underlying subspace were assumed to be the same. LSDA optimized the Fisher criterion and made further approximations on the inter-class scatter to make the optimization tractable; MMSDA optimized the max-min distance criterion, resulting in solving a series of time-consuming semi-definite programming problems and cannot scale to high dimension. Differently, in our method, the discovery of temporal structures is DTW-based and only depends on deterministic operations, which avoids the estimation of massive parameters of HMM; The latent alignments in the subspace can be jointly learned with the projection owing to our construction of separability.

Recurrent Neural Networks (RNNs) (Graves et al., 2013; Sutskever et al., 2014) have seldom been used for DR but often as classifiers. The sequences can be projected first by our method, and then input to RNNs for classification. This way, the input sequences are more discriminative and RNNs need to learn fewer parameters.

3. Latent Temporal Linear Discriminant Analysis

3.1. Learning Abstract Templates

We learn an abstract template $\mathbf{M}$ consisting of ordered temporal structures for each sequence class from all its training sequence samples. Each sequence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$ consists of a series of ordered frame-wide feature vectors, where $\mathbf{x}_t$ is the feature vector extracted from the t-th frame, and T is the length of the sequence. For a specific sequence class, we denote its training sequence sample set by $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N\}$, where N is the number of training sequences in the set, and T_n is the length of $\mathbf{X}_n$. Different sequence samples may have different lengths.

We define the abstract template as a sequence of the abstracted temporal structures $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_L]$, where the element $\mathbf{m}_j$ captures the average frame-wide features of a temporal structure or stage that each sequence must go through. Hence $\mathbf{M}$ can be considered as an atomic sequence. L is the length of $\mathbf{M}$, which is generally shorter than any sequence sample, because the learned template on-

ly contains the essential temporal structures and each structure will last several frames in a sequence.

$\mathbf{M}$ can be used to divide a sequence sample $\mathbf{X}$ into different temporal regions. This is achieved by aligning $\mathbf{X}$ to $\mathbf{M}$ with a warping function, which can be defined by a warping path $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_L]$. $\mathbf{p}_t = [s_t, e_t]^T$ means that the $\{s_t, s_t+1, \cdots, e_t\}$-th elements in $\mathbf{X}$ are aligned to the t-th element of $\mathbf{M}$. Similar to DTW, several constraints are applied to $\mathbf{P}$: (1) $s_1 = 1, e_L = T$; (2) $e_t < s_{t+1}, \forall t = 1, \cdots, L-1$; (3) $e_t \geq s_t$; (4) $l_t \leq a\frac{T}{L}$, where $l_t = e_t - s_t + 1$ is the number of elements in $\mathbf{X}$ that are aligned to the t-th element in $\mathbf{M}$, $a \geq 1$ is a factor that controls the allowed degree of warping. This constraint means that the number of elements in $\mathbf{X}$ aligned to any element in $\mathbf{M}$ should not exceed a multiple of the average number, and hence prevents extremely unbalanced partitioning. Therefore, only salient temporal structures that are universal in all training sequences can be captured by $\mathbf{M}$.

We employ the modified DTW algorithm (Su et al., 2016; 2017b) to compute the optimal warping path. We denote the cost of a partial path of aligning the first i elements in $\mathbf{X}$ to the first j elements in $\mathbf{M}$ as $c(i, j, l)$, where the last l elements of the first i elements in $\mathbf{X}$ are aligned to the j-th element in $\mathbf{M}$. $c(i, j, l)$ can be determined recurrently:

$$c(i,j,l) = \begin{cases} d(i,j), l = 1, i = j = 1 \\ d(i,j) + \min\limits_{k=1}^{aT/L} c(i-1, j-1, k), l = 1 \\ d(i,j) + c(i-1, j, l-1), l \leq a\frac{T}{L} \\ Inf, otherwise \end{cases}, \tag{1}$$

where $d(i, j)$ is the Euclidean distance between the i-th element of $\mathbf{X}$ and the j-th element of $\mathbf{M}$. The minimum alignment cost can be found by such a dynamic programming and is achieved at the end of recursion. The corresponding optimal warping path is obtained by back tracking.

Based on the dynamic alignment Eq. (1), $\mathbf{M}$ can be obtained by employing the *DTW barycenter averaging (DBA)* (Petitjean et al., 2011) as follows. We first use the uniform alignments to initialize $\mathbf{M}$. Specially, in the n-th training sequence $\mathbf{X}_n$, $l_j^n = \frac{T_n}{L}$ elements in $\mathbf{X}_n$ are aligned to the j-th element of $\mathbf{M}$, $\forall j = 1, \cdots, L$. The initial j-th element $\mathbf{m}_j$ of $\mathbf{M}$ can be computed as:

$$\mathbf{m}_j = \frac{1}{\sum\limits_{n=1}^{N} l_j^n} \sum_{n=1}^{N} \sum_{k=s_j^n}^{e_j^n} \mathbf{x}_k^n, \tag{2}$$

where $\mathbf{P}^n = [\mathbf{p}_1^n, \cdots, \mathbf{p}_L^n]$ is the alignment path that aligns $\mathbf{X}_n$ to $\mathbf{M}$, $\mathbf{p}_j^n = [s_j^n, e_j^n]^T$ records the start and end indexes of elements in $\mathbf{X}_n$ that are aligned to $\mathbf{m}_j$. We then align each training sequence $\mathbf{X}_n$ to the initial $\mathbf{M}$ using Eq. (1) to update the alignment path $\mathbf{P}^n$, for $n = 1, \cdots, N$. We finally recompute the elements in $\mathbf{M}$ using Eq. (2) with the updated $\mathbf{P}^n$ again. This process can be repeated until the difference of $\mathbf{M}$ in the current iteration and $\mathbf{M}$ in the previous iteration is below a threshold or a maximum number of iterations is reached. We summarize the abstract template learning algorithm in Alg. 1.

Algorithm 1 Abstract template learning

Input: $\{\mathbf{X}_1, \cdots, \mathbf{X}_N\}$; L; a;
Output: $\mathbf{M}$; $\mathbf{P}^n, n = 1, \cdots, N$;

1: Initialize the uniform alignment path $\mathbf{P}^n$ for the training sequence $\mathbf{X}_n$, for $n = 1, \cdots, N$;
2: Compute the initial abstract template $\mathbf{M}$ using Eq. (2);
3: **while** $\mathbf{M}$ has not converged **do**
4: Update the alignment paths $\mathbf{P}^n$ by aligning $\mathbf{X}_n$ to $\mathbf{M}$, $n = 1, \cdots, N$ using Eq. (1);
5: Update the abstract template $\mathbf{M}$ with the alignment paths $\mathbf{P}^n$, $n = 1, \cdots, N$ using Eq. (2);
6: **end while**

Alg. 1 extends DBA to multi-dimensional sequences with a uniform initialization, and imposes stricter constraints on the warping path. As a result, any vector in any sequence can only be aligned to one element of $\mathbf{M}$, which facilitates the invariant property of the separability in Sec. 3.2.

Convergence. Alg. 1 actually minimizes the following objective function:

$$\min_{\mathbf{P}^n, n=1,\cdots,N} \sum_{j=1}^{L} \sum_{n=1}^{N} \sum_{k=s_j^n}^{e_j^n} \|\mathbf{x}_k^n - \mathbf{m}_j\|_2^2. \tag{3}$$

The value of the objective function in Eq.(3) decreases by both alternative procedures in Alg.1. The objective function also has a lower bound 0. Thus Alg.1 is guaranteed to converge to a local minimum.

3.2. Separability Construction

We measure the separability among sequence classes based on their abstract templates in two aspects: the within-class scatter and the inter-class distance. We define the intra-class scatter of a sequence class as the sum of variances of all component temporal structures in the abstract template:

$$\mathbf{S} = \sum_{j=1}^{L} (\sum_{n=1}^{N} l_j^n / \sum_{n=1}^{N} T_n) \mathbf{S}_j. \tag{4}$$

l_j^n is the number of features in the n-th sequence aligned to the j-th temporal structure in the abstract template. $\mathbf{S}_j$ is the variance of the j-th temporal structure, which can be estimated as the variance matrix of all feature vectors in all training sequences aligned to the j-th element of $\mathbf{M}$.

For sequence class i, we denote its intra-class scatter by $\mathbf{S}^i$. Assuming there are C sequence classes, we define the within-class scatter as the sum of intra-class scatters of all classes weighted by the prior probability p^i of class i:

$$\mathbf{S}_w = \sum_{i=1}^{C} p^i \mathbf{S}^i, \tag{5}$$

where p^i can be estimated as the number of sequences from class i divided by the number of sequences from all classes.

The learned abstract template $\mathbf{M}$ of a sequence class represents the temporal structures and their general evolution of the class. The separability between two sequence classes can be reflected by the difference between the two corresponding abstract templates. For two sequence classes i and j, we define the separability between them as:

$$\mathbf{S}_b = \sum_{1\le i<j\le C} \sum_{1\le u,v\le L} p_u^i p_v^j (\mathbf{m}_u^i - \mathbf{m}_v^j)(\mathbf{m}_u^i - \mathbf{m}_v^j)^T. \tag{6}$$

$\mathbf{m}_u^i$ and $\mathbf{m}_v^j$ denote the u-th element of $\mathbf{M}^i$ and the v-th element of $\mathbf{M}^j$, respectively. p_u^i and p_v^j denote the prior probabilities of $\mathbf{m}_u^i$ and $\mathbf{m}_v^j$, respectively. p_u^i is estimated as the number of vectors in sequences from class i that are aligned to $\mathbf{m}_u^i$ divided by the number of all vectors in all sequences from all classes.

Constructing the inter-class scatter by Eq. (6) is equivalent to viewing each temporal structure as a subclass. Since each sequence class is abstracted by several ordered temporal structures, if all temporal structures from all classes are maximally separated, the separability of different sequence classes increases accordingly. Thus Eq. (6) can indeed reflect the separability between sequence classes.

Note that both $\mathbf{S}_w$ (5) and $\mathbf{S}_b$ (6) rely on the alignments of sequence samples to the corresponding abstract templates: $\mathbf{P} = \{\mathbf{P}_n^i, n = 1, 2, \cdots, N^i, i = 1, 2, \cdots, C\}$. We denote them by $\mathbf{S}_w(\mathbf{P})$ and $\mathbf{S}_b(\mathbf{P})$[1], respectively, to emphasize the dependencies on alignments.

Compared with HMM-based separability (Su & Ding, 2013; Su et al., 2018), our separability construction has several advantages. (1). It does not require a large amount of training data. Even when each class has only one sequence sample, Alg. 1 can still be performed and meaningful scatters can thereby be constructed. In this case, Alg. 1 degrades to the temporal clustering algorithm (Su et al., 2016). (2). It does not need to estimate any parameter, thus has better scalability. (3). Owing to the constraints on the warping path, calculating $\mathbf{S}_w$ by Eq. (5) is also equivalent to viewing all temporal structures in all classes as subclasses. Thus $\mathbf{S}_b(\mathbf{P}) + \mathbf{S}_w(\mathbf{P}) = \mathbf{S}_t$, where $\mathbf{S}_t$ is the total scatter of all features in all sequences and is independent of $\mathbf{P}$. This invariant property ensures the joint optimization in Sec. 3.3.

[1]Strictly speaking, they also depend on $\mathbf{M}$, but $\mathbf{M}$ and $\mathbf{P}$ are closely associated, so we omit $\mathbf{M}$ for brevity.

3.3. Joint Learning of the Transformation and the Latent Alignments

Our goal is to learn a linear transformation matrix $\mathbf{W} \in \mathbb{R}^{\mathbf{d}\times\mathbf{d}'}$ to project feature vectors in sequences from the original d-dimensional space to the most discriminative d'-dimensional subspace, in which the separability among different sequence classes are maximized. The separability depends on the alignments between the sequences and the abstract templates, which are inferred based on the pairwise distances between feature vectors in the space. When the features are projected to a subspace, the distances among the transformed features may change. The alignments may change accordingly, which should be re-calculated using Alg. 1 in the subspace. The updates of the alignments in turn affect the determination of the transformation. Existing methods (Su & Ding, 2013; Su et al., 2018) tackle such entanglement by fixing the alignments obtained in the original space, which may lead to sub-optimal solutions.

We consider the joint learning of the transformation and the abstract templates together with the corresponding temporal alignments in the latent subspace simultaneously. We optimize the Fisher criterion that maximizes the inter-class separability and minimizes the within-class scatter. Due to the invariant property: $\mathbf{S}_b(\mathbf{P}) + \mathbf{S}_w(\mathbf{P}) = \mathbf{S}_t$, the optimal projections of maximizing the ratio of $\mathbf{S}_b$ and $\mathbf{S}_w$ and maximizing the ration of $\mathbf{S}_b$ and $\mathbf{S}_t$ are the same (Fukunaga, 1990). Therefore, we formulate our objective function as follows:

$$\max_{\mathbf{W},\mathbf{P}} tr((\mathbf{W}^T\mathbf{S}_t\mathbf{W})^{-1}\mathbf{W}^T\mathbf{S}_b(\mathbf{P})\mathbf{W}). \tag{7}$$

We solve Eq. (7) by alternatively updating $\mathbf{W}$ and $\mathbf{P}$ to obtain a local optimal solution. We call this method *Latent Temporal Linear Discriminant Analysis* or LT-LDA, which is summarized in Alg. 2.

In the first stage, LT-LDA optimizes over $\mathbf{P}$ by fixing $\mathbf{W}$. The first inverse matrix item $(\mathbf{W}^T\mathbf{S}_t\mathbf{W})^{-1}$ in Eq. (7) does not depend on $\mathbf{P}$. We omit this item for the moment to derive an intuitive solution and will explain its effect later. The objective then becomes

$$\max_{\mathbf{P}} tr(\mathbf{W}^T\mathbf{S}_b(\mathbf{P})\mathbf{W}). \tag{8}$$

Since $\mathbf{S}_b(\mathbf{P}) = \mathbf{S}_t - \mathbf{S}_w(\mathbf{P})$, Eq. (8) is equivalent to:

$$\min_{\mathbf{P}} tr(\mathbf{W}^T\mathbf{S}_w(\mathbf{P})\mathbf{W}). \tag{9}$$

Substituting Eq. (5) to Eq. (9) and expanding, Eq. (9) is

Algorithm 2 LT-LDA

Input: the training sequences of each class $c = 1, \cdots, C$, the length of the abstract template L, the control factor a;
Output: the projection $\mathbf{W}$;

1: Initialize the abstract template $\mathbf{M}^c$ and the associated alignments $\mathbf{P}^c$ in the original space using Alg. 1, for $c = 1, \cdots, C$; Calculate $\mathbf{S}_w$ (5) and $\mathbf{S}_b$ (6) according to $\mathbf{P}^c$ and $\mathbf{M}^c$;
2: Initialize $\mathbf{W}$ by solving Eq.(11)
3: **while** $\mathbf{W}$ has not converged **do**
4: Project the training sequences into a subspace by $\mathbf{W}$; Update $\mathbf{M}^c$ and $\mathbf{P}^c$ in this subspace using Alg. 1, for $c = 1, \cdots, C$;
5: Re-calculate $\mathbf{S}_w$ and $\mathbf{S}_b$ with the updated alignments $\mathbf{P}$ by Eq. (5) and Eq. (6), respectively;
6: Update $\mathbf{W}$ by solving Eq. (11);
7: **end while**

transformed to:

$$\sum_{i=1}^{C} p^i \min_{\mathbf{P}^{in}, n=1,\cdots,N} \sum_{j=1}^{L} \sum_{n=1}^{N^i} \sum_{k=s_j^{in}}^{e_j^{in}} \left\| \hat{\mathbf{x}}_k^{in} - \hat{\mathbf{m}}_j^i \right\|_2^2, \quad (10)$$

where $\hat{\mathbf{x}}_k^{in} = \mathbf{W}^T \mathbf{x}_k^{in}$ and $\hat{\mathbf{m}}_j^i$ are the projected feature and the element of the abstract template in the subspace, respectively. The superscript i is to indicate that the variable belongs to class i. To ensure the convergence and compensate the omitted item when deriving Eq. (8) which will be more clear in Section 3.4, the features should first be centered before the start of the iterations, and a whitening preprocessing should be applied to all features in all sequences in this stage. That is, the mean of all $\mathbf{x}_k^{in}$ is zero, and $\hat{\mathbf{x}}_k^{in} = \mathbf{W}_w \mathbf{W}^T \mathbf{x}_k^{in}$, where $\mathbf{W}_w = \mathbf{\Gamma}_w^{-\frac{1}{2}}$ is whitening transformation and $\mathbf{\Gamma}_w$ is the total scatter of all projected features in all sequences. In our experiments, we found that the two procedures can be neglected, and the LT-LDA still converges while the computational complexity is reduced.

Each of the C components of minimization is exactly the same with Eq. (3) in the subspace associated with $\mathbf{W}$ instead of the original space. These minimizations are independent from each other, and hence we can learn the abstract template and the corresponding alignments of training sequences for each of the C classes using Alg. 1 individually. The learned alignments for all the sequences in all the classes are used to update $\mathbf{S}_w$ and $\mathbf{S}_b$ using Eq. (5) and Eq. (6), respectively.

In the second stage, LT-LDA optimizes over $\mathbf{W}$ for given $\mathbf{P}$. In this case, both $\mathbf{S}_w$ and $\mathbf{S}_b$ are fixed, and the objective function becomes a standard LDA problem:

$$\begin{aligned} &\max_{\mathbf{W}} tr((\mathbf{W}^T \mathbf{S}_t \mathbf{W})^{-1} \mathbf{W}^T \mathbf{S}_b \mathbf{W}) \\ \Leftrightarrow &\max_{\mathbf{W}} tr((\mathbf{W}^T \mathbf{S}_w \mathbf{W})^{-1} \mathbf{W}^T \mathbf{S}_b \mathbf{W}). \end{aligned} \quad (11)$$

The columns of the updated $\mathbf{W}$ are given by the eigenvectors of $\mathbf{S}_w^{-1} \mathbf{S}_b$ with respect to the d' largest eigenvalues.

3.4. Theoretical Analysis

We theoretically provide more insights by proving 1) that the abstract template learning algorithm (Alg. 1) can be linked to a trace maximization formulation; 2) that the LT-LDA algorithm (Alg. 2) is guaranteed the converge; 3) that it is possible to simplify the joint optimization of Eq. (13) under certain conditions. All proofs are given in the supplementary material.

Let $\mathbf{Z}$ be the matrix consisting of all frame-wise feature vectors in all training sequences. Let $\mathbf{T}$ be the alignment indicator matrix, which is defined as follows: $\mathbf{T} = \{\pi_{i,k}\}_{N_t \times CL}$, where $\pi_{i,k} = 1$ if the frame-wide feature vector $\mathbf{z}_i$ in the i-th column of $\mathbf{Z}$ is in the sequence from the $c = \lceil (k - \frac{1}{2})/L \rceil$-th class and is aligned to the $l = k - (c-1)L$-th stage of class c, and $\pi_{i,l} = 0$ otherwise. N_t is the total number of vectors in all the training sequences from all the samples. Following (Dhillon et al., 2005; Ye et al., 2007), the weighted indicator matrix is defined as $\mathbf{F} = \mathbf{T}(\mathbf{T}^T \mathbf{T})^{-\frac{1}{2}}$.

It can be shown that

$$\mathbf{F}_{i,k} = \begin{cases} 1/\sqrt{n_k}, if\ \mathbf{z}_i \in (c,l) \\ 0, otherwise \end{cases},$$

where n_k is the number of 1 in the k-th column of $\mathbf{F}$, *i.e.*, the number of vectors that are aligned to stage l of class c.

Lemma 1. *Objective function (3) is equivalent to the trace maximization problem*

$$\max_{\mathbf{F}} tr(\mathbf{F}^T \mathbf{Z}^T \mathbf{Z} \mathbf{F}). \quad (12)$$

Lemma 2. *Objective function (7) is equivalent to the trace maximization problem*

$$\max_{\mathbf{W},\mathbf{F}} tr(\mathbf{F}^T \mathbf{Z}^T \mathbf{W} (\mathbf{W}^T \mathbf{Z} \mathbf{Z}^T \mathbf{W})^{-1} \mathbf{W}^T \mathbf{Z} \mathbf{F}). \quad (13)$$

Theorem 1. *The LT-LDA algorithm (Alg. 2) is guaranteed the converge.*

Similar to (Ye et al., 2007), in some specific cases, the joint optimization of Eq. (13) can be simplified by factoring out the projection matrix $\mathbf{W}$. The result is summarized as follows:

Theorem 2. *Let $\mathbf{G} = \mathbf{Z}^T \mathbf{Z}$ be the Gram matrix. When the dimensionality is reduced to a specific value $d' = \min(CL, d, N_t)$ and a regularization term $\delta \mathbf{I}_{N_t}$ is added to the total scatter $\mathbf{S}_t$, where $\mathbf{I}_{N_t}$ is the N_t-order identity matrix, if $\mathbf{W}^*$ and $\mathbf{F}^*$ are the optimal solutions of the trace maximization problem (13):*

$$\max_{\mathbf{W},\mathbf{F}} tr(\mathbf{F}^T \mathbf{Z}^T \mathbf{W} (\mathbf{W}^T (\mathbf{Z}\mathbf{Z}^T + \delta \mathbf{I}_{N_t}) \mathbf{W})^{-1} \mathbf{W}^T \mathbf{Z} \mathbf{F}) \quad (14)$$

then $\mathbf{F}^*$ *is also the optimal solution of the problem*

$$\max_{\mathbf{F}} tr(\mathbf{F}^T(\mathbf{I}_{N_t} - (\mathbf{I}_{N_t} + \frac{1}{\delta}\mathbf{G})^{-1})\mathbf{F}) \qquad (15)$$

This theorem provides an upper bound of the objective for the stage of learning the partitions, which provides additional insights on the subspace selection of LT-LDA. From Lemma 1, in the original space, the objective function (12) of the abstract template leaning actually maximizes $tr(\mathbf{F}^T\mathbf{Z}^T\mathbf{Z}\mathbf{F}) = tr(\mathbf{F}^T\mathbf{G}\mathbf{F})$. While in the d'-dimensional subspace, the objective (15) of LT-LDA actually maximizes a kernel version of Eq. (12), where the kernel Gram matrix $\mathbf{G}_k = \mathbf{I}_{N_t} - (\mathbf{I}_{N_t} + \frac{1}{\delta}\mathbf{G})^{-1}$ is used instead of the original Gram matrix $\mathbf{G}$ in Eq. (12). $\mathbf{G}_k \to \mathbf{G}/\delta$ when $\delta \to \infty$, and hence objective (15) is equivalent to standard objective (12). $\mathbf{G}_k \to \mathbf{U}_r\mathbf{U}_r^T$ when $\delta \to 0$, $\mathbf{U}_r$ is the set of the largest r principal components of all the features in all the sequences *w.r.t.* the non-zero eigenvalues of $\mathbf{G}$. Thus objective (15) is equivalent to learning the abstract templates in the subspace determined by PCA.

$\mathbf{G}_k$ can be further expressed as

$$\mathbf{G}_k = \mathbf{U}diag(\lambda_1/(\lambda_1+\delta), \cdots, \lambda_{N_t}/(\lambda_{N_t}+\delta))\mathbf{U}^T.$$

This means that, the iterative procedures of LT-LDA essentially construct a kernel matrix for learning the latent alignments w.r.t. the abstract templates. The construction is achieved by performing a transformation to $\mathbf{G}$, such that each eigenvalue λ of $\mathbf{G}$ is transformed to $\lambda/(\lambda+\delta)$, while the eigenvectors of $\mathbf{G}$ remain unchanged. The subspace can be determined easily given the alignments without the need of the iterative procedures. The nature of the subspace selection by LT-LDA indicates that it may be possible to accelerate the LT-LDA algorithm by fixing the partitions learned by optimizing (15) and hence getting rid of the time-consuming iterative procedures, without significant degradation in performance.

3.5. Computational Complexity

The complexity of updating $\mathbf{P}$ using Alg. 1 for all the C classes is $O(ICNLTd)$, I is the number of iterations in Alg. 1.. The complexity of re-calculating $\mathbf{S}_w$, $\mathbf{S}_b$ and $\mathbf{W}$ is $O(CNTd^2 + C^2L^2d^2 + d^3)$. Thus the overall complexity of Alg. 2 is $O(I'(ICNLTd + CNTd^2 + C^2L^2d^2 + d^3))$, I' is the number of iterations in Alg. 2.

4. Experimental Results

In this section we evaluate the proposed LT-LDA in comparison with several supervised DR methods for sequences on three real-world datasets. Evaluations on another dataset are presented in the supplementary material.

4.1. Experimental Setup

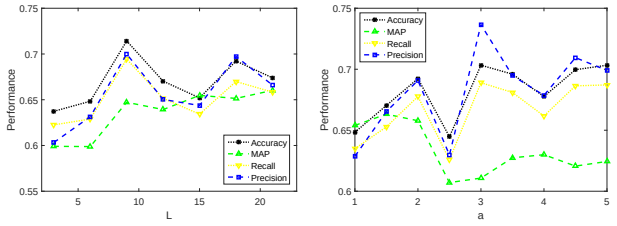
Datasets. **ChaLearn Gesture dataset (Escalera et al., 2013b;a)** contains Kinect videos from 20 Italian gestures. The dataset has been split into training, validation and test sets. **MSR Sports Action3D dataset (Li et al., 2010)** consists of depth sequences from 20 sports actions. We follow the same experimental setup as in (Wang et al., 2012; Wang & Wu, 2013) to split the dataset into training and test set. **Olympic Sports dataset (Niebles et al., 2010)** consists of 783 video sequences from 16 actions. The dataset has been split into training and test sets, where 649 videos are used for training and 134 videos are used for testing.

Frame-wise features. We extract a feature vector from each frame, and hence every action video is represented by a sequence of frame-wise features. For the Chalearn dataset, we employ the frame-wise features provided by the authors of (Fernando et al., 2015), which are body-joints-based features with a dimensionality of 100. For the MSR Action3D dataset, we employ the frame-wise features provided by the authors of (Wang & Wu, 2013), which are the relative positions of all the 3D joints with a dimensionality of 192. For the Olympic Sports dataset, we employ the improved dense trajectories (Wang & Schmid, 2013) based frame-wise features. MBH descriptors are extracted at densely sampled points from each frame and then encoded by Bag-of-Words with a codebook of $4,000$ visual words. The frame-wise feature is the histogram of the quantized descriptors with a dimensionality of $4,000$.

Classification and evaluation measures. We adopt three classifiers in the learned subspace, including the HMM classifier, the DTW classifier, and the SVM classifier. For the HMM classifier, a left-to-right HMM with 4 states and self-loops is trained for each sequence class, and a test sequence is classified to the class whose HMM has the highest probability to generate it. For the DTW classifier, the training sequence that has the smallest sum of DTW distances with all other sequences from the same class is selected as the template of this class. A test sequence is classified to the class whose temple has the smallest DTW distance to it. The two classifiers directly take sequences as input, and we use the accuracy as the performance measure. For the SVM classifier, we encode each sequence into a vector by rank pooling (Fernando et al., 2015). Linear SVMs are trained on these encoded vectors, and the parameter C of SVM is selected by cross-validation. We use the accuracy and the Mean Average Precision (MAP) as the evaluation measures for the SVM classifier.

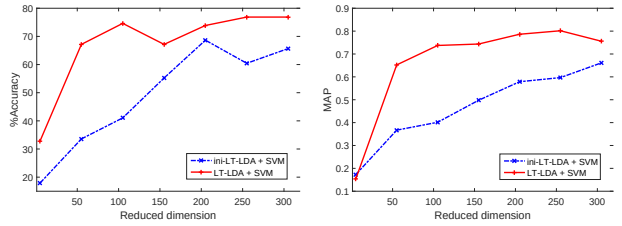
4.2. Influence of Parameters

The proposed LT-LDA has two preset parameters: the length of each abstract template L and the factor a controlling the allowed degree of warping. In this section we eval-



(a) (b)

Figure 1. Different performances with the SVM classifier as functions of (a) the length L of the abstract template and (b) the control factor a on the MSR Action3D dataset.



(a) (b)

Figure 2. Comparisons of the proposed LT-LDA without and with the joint learning of the latent alignments. (a) Accuracies and (b) MAPs with the SVM classifier as functions of the dimensionality of the subspace on the Olympic Sports dataset.

uate the influence of them on the MSR Action3D dataset. Evaluations on other datasets are presented in the supplementary material. Different performance measures including accuracy, MAP, precision and recall with the SVM classifier are evaluated by increasing L from 3 to 21 with an interval of 3 while fixing a to 2, and increasing a from 1 to 5 with an interval of 0.5 while fixing L to 8, respectively. The reduced dimension is fixed to 20. The results are shown in Fig. 1. The optimal parameters are generally the same for multi-class indicators including accuracy, precision, recall and F-score, but are different for MAP.

LT-LDA achieves the highest multi-class performances when $L = 9$ on this dataset. The larger the L, the longer the template, the finer the captured temporal structures, but the less accurate the estimated statistics of structures, and the more likely to cause overfitting. Therefore, the performances decrease if the length L is too long or too short. Generally, setting L within the range of 6 to 9 leads to satisfactory results. A too large a easily leads to unbalanced alignments. If a is too small, the flexibility of alignments may be restricted. Allowing appropriate warping leads to satisfactory results. We fix a to 2 in the following experiments, and fix L to 8 except on the Olympic Sports dataset, where we set L to 20 such that LT-LDA can preserve $20 \times C - 1 = 319$ dimensions at most.

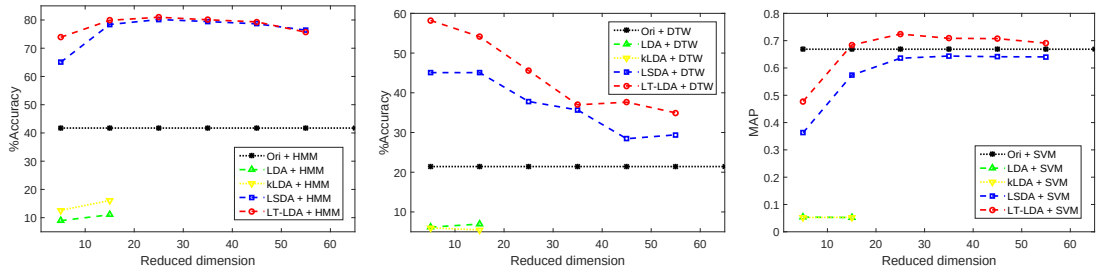
4.3. Effects of the Joint Learning

In LT-LDA, the latent alignments are jointly learned with the underlying subspace. If instead we use the alignments in the original space calculated by Alg. 1 directly, LT-LDA degenerates to the initialization of $\mathbf{W}$ in LT-LDA. We denote this algorithm by ini-LT-LDA and compare it with LT-LDA on the large-scale Olympic Sports dataset. The comparisons by using different classifiers and evaluation measures are shown in Fig. 2. We can observe that LT-LDA significantly outperforms ini-LT-LDA by a large margin. Learning the latent alignments associated with the subspace jointly does help to improve the classification performance in the subspace. This is because the temporal structures and the alignments may change from those in the original space. In the learned subspace of ini-LT-LDA, although different classes get better separated under the alignments in the original space, additional confusions may be introduced due to the changes of alignments. While for LT-LDA, since the separability is maximized in the subspace under the corresponding alignments, the learned subspace gets joint optimality among all possible subspaces. More complete evaluations and analysis on more datasets are presented in the supplementary material.

4.4. Comparison with Different DR Methods

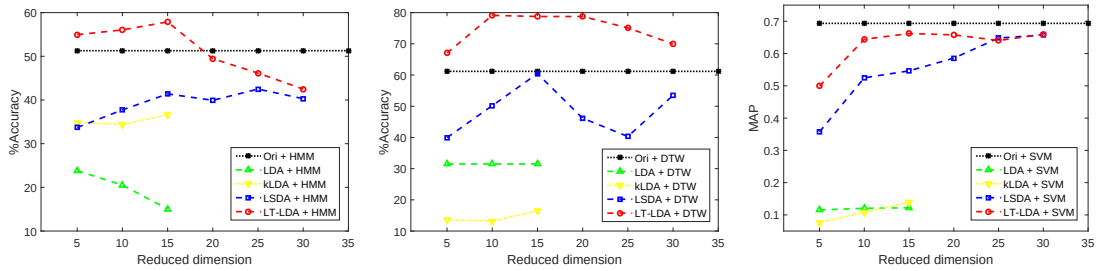
We compare the proposed LT-LDA with LDA and kernel LDA (kLDA) by viewing the features in sequences as independent samples, as well as LSDA. The performances of the original feature sequences are also presented as baselines. We use the drtoolbox (van der Maaten & Hinton, 2008) to perform LDA and kLDA. For kLDA, it is impracticable to use all features in all training sequences, because this will lead to a huge size of the kernel matrix and very large space and computational overhead. Following (Su et al., 2018), we sample 1 to 5 features randomly from each sequence for training. We use the same parameters of LSDA as in (Su & Ding, 2013).

Fig. 3 and Fig. 4 depict the performances as functions of the dimensionality of the learned subspace on the ChaLearn dataset and the Action3D dataset, respectively. We can observe that the proposed LT-LDA achieves the best performances among all these DR methods by all the three classifiers with different evaluation measures on both datasets. Especially by the DTW classier, LT-LDA outperforms the second LSDA by a margin of more than 10%. By the HMM classifier and the DTW classifier, the accuracies of LT-LDA are consistently better on all the reduced dimensions, and LT-LDA with less than 15 dimensions achieves much better results than the original features with hundreds of dimensions. For the SVM classifier with the rank pooling, LT-LDA achieves comparable MAPs with original features using only 15 or 25 dimensions. The worse performances of LDA and kLDA are caused by the dependency of features in sequences, which violates the basic assumption of



(a) (b) (c)

Figure 3. (a) Accuracies with the HMM classifier (b) accuracies with the DTW classifier and (c) MAPs with the rank pooling and the SVM classifier as functions of the dimensionality of the subspace on the Chalearn Gesture dataset.



(a) (b) (c)

Figure 4. (a) Accuracies with the HMM classifier (b) accuracies with the DTW classifier and (c) MAPs with the rank pooling and the SVM classifier as functions of the dimensionality of the subspace on the Action3D dataset.

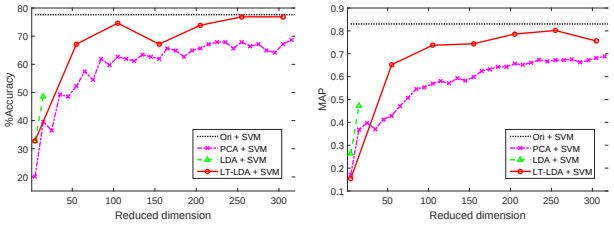
the two methods, while LT-LDA well exploits such temporal dependencies by learning the latent alignments.

For the Olympic Sports dataset, there are only less than 35 training videos per class. Each video generally has hundreds of frames, and the dimensionality of the feature for each frame is $4,000$. Therefore, it is impracticable to train a HMM for each class and hence LSDA cannot be employed. kLDA is also computational prohibited. We compare LT-LDA with PCA and LDA on this dataset, as shown in Fig. 5. LDA can only preserve $C-1=19$ dimensions at most. LT-LDA consistently outperforms PCA, and further improves the performances when more than 19 dimensions are preserved. With only 250 dimensions, LT-LDA achieves comparable accuracy and MAP with the original BoW-based distributed features with $4,000$ dimensions. This implies that the BoW features can be greatly compressed by LT-LDA while the discriminative information is maintained.

To compare with the state-of-the-art gesture recognition methods, we also evaluate the multi-class precision, recall, and F-score by LT-LDA with fine-tuned a via the rank pooling and the SVM classifier on the ChaLearn dataset. The comparisons are shown in Tab. 1. LT-LDA outperforms the state-of-the-art results using only 45 dimensions.

5. Conclusion

In this paper, we have presented a DR method for sequence data, called LT-LDA, which learns the subspace and infers



(a) (b)

Figure 5. (a) Accuracies and (b) MAPs with the rank pooling and the SVM classifier as functions of the dimensionality of the subspace on the Olympic Sports dataset.

Table 1. Comparison with state-of-the-art results on the ChaLearn dataset.

Method	Precision	Recall	F-score
(Wu et al., 2013)	0.599	0.593	0.596
(Pfister et al., 2014)	0.612	0.623	0.617
(Fernando et al., 2017)	0.753	0.751	0.752
(Su et al., 2018)	0.768	0.767	0.767
LT-LDA+SVM	**0.784**	**0.783**	**0.783**

the latent alignments within it simultaneously. We formulate the learning of the subspace, the latent alignments, and the temporal structures into a joint objective function, and solve it by iteratively repeating the two alternative procedures of applying LDA and learning the abstract templates. The effectiveness of the proposed method is demonstrated on three action datasets with various evaluation measures and classifiers.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No.61603373, the National Science Foundation grant IIS-1217302, IIS-1619078, and the Army Research Office ARO W911NF-16-1-0138.

References

Bian, W. and Tao, D. Max-min distance analysis by using sequential sdp relaxation for dimension reduction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(5): 1037–1050, 2011.

De la Torre, F. and Kanade, T. Discriminative cluster analysis. In *Proc. IEEE Int'l Conf. Machine Learning*, 2006.

Dhillon, I., Guan, Y., and Kulis, B. A unified view of kernel k-means. spectral clustering and graph cuts. *Technical report, Department of Computer Sciences, University of Texas at Austin*, 2005.

Ding, C. and Li, T. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proc. IEEE Int'l Conf. Machine Learning*, 2007.

Escalera, S., Gonzàlez, J., Baró, X., Reyes, M., Guyon, I., Athitsos, V., Escalante, H., Sigal, L., Argyros, A., Sminchisescu, C., et al. Chalearn multi-modal gesture recognition 2013: grand challenge and workshop summary. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 365–368. ACM, 2013a.

Escalera, S., Gonzàlez, J., Baró, X., Reyes, M., Lopes, O., Guyon, I., Athitsos, V., and Escalante, H. Multi-modal gesture recognition challenge 2013: Dataset and results. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 445–452. ACM, 2013b.

Fernando, B., Gavves, E., M., J. O., Ghodrati, A., and Tuytelaars, T. Modeling video evolution for action recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2015.

Fernando, B., Gavves, E., M., J. O., Ghodrati, A., and Tuytelaars, T. Rank pooling for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):773–787, 2017.

Fisher, R. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

Flamary, R., Tuia, D., Labb, B., Camps-Valls, G., and Rakotomamonjy, A. Large margin filtering. *IEEE Transactions on Signal Processing*, 60(2):648–659, 2012.

Fukunaga, K. *Introduction to statistical pattern recognition*. NewYork: Academic Press, 1990.

Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pp. 6645–6649. IEEE, 2013.

Lajugie, R., Garreau, D., Bach, F., and Arlot, S. Metric learning for temporal sequence alignment. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1817–1825, 2014.

Li, W., Zhang, Z., and Liu, Z. Action recognition based on a bag of 3d points. In *IEEE Int'l Workshop on CVPR for Human Communicative Behavior Analysis*, 2010.

Niebles, J. C., Chen, C.-W., and Fei-Fei, L. Modeling temporal structure of decomposable motion segments for activity classification. In *European conference on computer vision*, pp. 392–405. Springer, 2010.

Nikitidis, S., Zafeiriou, S., and Pantic, M. Merging svms with linear discriminant analysis: A combined model. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2014.

Petitjean, F., Ketterlin, A., and Gançarski, P. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

Pfister, T., Charles, J., and Zisserman, A. Domain-adaptive discriminative one-shot learning of gestures. In *European Conference on Computer Vision*, pp. 814–829. Springer, 2014.

Rabiner, L. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Sakoe, H. and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978.

Shyr, A., Urtasun, R., and Jordan, M. I. Sufficient dimension reduction for visual sequence classification. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 3610–3617, 2010.

Su, B. and Ding, X. Linear sequence discriminant analysis: a model-based dimensionality reduction method for vector sequences. In *Proc. IEEE Int'l Conf. Computer Vision*, pp. 889–896, 2013.

Su, B., Zhou, J., Ding, X., Wang, H., and Wu, Y. Hierarchical dynamic parsing and encoding for action recognition.

In *European Conference on Computer Vision*, pp. 202–217. Springer, 2016.

Su, B., Ding, X., Liu, C., Wang, H., and Wu, Y. Discriminative transformation for multi-dimensional temporal sequences. *IEEE Trans. Image Processing*, 26(7):3579–3593, 2017a.

Su, B., Zhou, J., Ding, X., and Wu, Y. Unsupervised hierarchical dynamic parsing and encoding for action recognition. *IEEE Trans. Image Processing*, 26(12):1057–7149, 2017b.

Su, B., Ding, X., Wang, H., and Wu, Y. Discriminative dimensionality reduction for multi-dimensional sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(1):77–91, 2018.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in Neural Inform. Process. Syst.*, pp. 3104–3112, 2014.

Trigeorgis, G., Nicolaou, M. A., Schuller, B. W., and Zafeiriou, S. Deep canonical time warping for simultaneous alignment and representation learning of sequences. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1128–1138, 2018.

van der Maaten, L. and Hinton, G. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Wang, H. and Schmid, C. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3551–3558, 2013.

Wang, J. and Wu, Y. Learning maximum margin temporal warping for action recognition. In *Proc. IEEE Int'l Conf. Computer Vision*, 2013.

Wang, J., Liu, Z., and Wu, Y. Mining actionlet ensemble for action recognition with depth cameras. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2012.

Wu, J., Cheng, J., Zhao, C., and Lu, H. Fusing multi-modal features for gesture recognition. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 453–460. ACM, 2013.

Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., and Lin, S. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.

Ye, J., Zhao, Z., and Wu, M. Discriminative k-means for clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

Zhang, Y. and Yeung, D.-Y. Worst-case linear discriminant analysis. In *Proc.Advances in Neural Information Processing Systems*, 2010.

Zhou, F. and De la Torre, F. Generalized time warping for multi-modal alignment of human motion. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 1282–1289, 2012.