Sufficient Dimension Reduction for Visual Sequence Classification

Alex Shyr
UC Berkeley
xshyr@eecs.berkeley.edu

Raquel Urtasun
TTI, Chicago
rurtasun@ttic.edu

Michael I. Jordan
UC Berkeley
jordan@eecs.berkeley.edu

Abstract

When classifying high-dimensional sequence data, traditional methods (e.g., HMMs, CRFs) may require large amounts of training data to avoid overfitting. In such cases dimensionality reduction can be employed to find a low-dimensional representation on which classification can be done more efficiently. Existing methods for supervised dimensionality reduction often presume that the data is densely sampled so that a neighborhood graph structure can be formed, or that the data arises from a known distribution. Sufficient dimension reduction techniques aim to find a low dimensional representation such that the remaining degrees of freedom become conditionally independent of the output values. In this paper we develop a novel sequence kernel dimension reduction approach (S-KDR). Our approach does not make strong assumptions on the distribution of the input data. Spatial, temporal and periodic information is combined in a principled manner, and an optimal manifold is learned for the end-task. We demonstrate the effectiveness of our approach on several tasks involving the discrimination of human gesture and motion categories, as well as on a database of dynamic textures.

1. Introduction

Many computer vision problems involve high dimensional datasets that are computationally challenging to analyze. In such cases it is desirable to reduce the dimensionality of the data while preserving the original information in the data distribution, allowing for more efficient learning and inference. Linear (e.g., PCA) and non-linear (e.g., LLE [14], Isomap [15], GPLVM [8]) unsupervised learning techniques learn a low dimensional space that represents "well" the data without regard to any particular task. Supervised dimensionality reduction approaches (e.g., Linear Discriminant Analysis [4], Discriminative GPLVM [17]) try to estimate a low-dimensional representation which has sufficient information for predicting the task target values. However, these supervised approaches assume that the latent space and/or the data is generated from some restricted distribution (e.g., a Gaussian process for the GPLVM). When the data do not follow this distribution, the bias introduced by this assumption can significantly affect performance.

Sufficient dimension reduction (SDR) techniques [9] aim to find a low-dimensional space such that vectors in its orthogonal complement become conditionally independent of the output values. Fukumizu et al. [5] proposed *kernel dimension reduction* (KDR) which unlike other SDR techniques does not make strong assumptions on the distribution of the input data. This is important for us, since data in computer vision applications rarely satisfy these assumptions. KDR makes use of cross-covariance operators which are infinite-dimensional generalizations of covariance matrices. Nonlinear dependencies can be captured by defining the cross-covariance operators on reproducing kernel Hilbert spaces (RKHSs).

However, to date, SDR has only been applied to static data, and has not been applied to common vision problems beyond learning a simple image manifold [13]. In this paper we extend KDR to model time-series data and design kernels that capture dynamics, periodic motions and multiclass classification. Our approach combines spatial, temporal and periodic information in a principled manner, and learns an optimal manifold without assuming any distribution of the data. In particular, we propose two ways of combining this information: multiple kernel learning and building regularizers that exploit the manifold structure of the dynamics.

We demonstrate the effectiveness of our approach on classifying human gestures and activities from video, motion capture data and dynamic textures with large intra-category variations. Our approach is shown to be superior to unsupervised methods (i.e., PCA), to classifying in the observation space using NN and SVMs, structure prediction using SVM-HMM [1], the original KDR [5], and sequence classification methods such as HMMs, CRFs [7] and HCRFs [19]. In the remainder of the paper, we first introduce our framework for sufficient dimension reduction of sequence data, present our experimental evaluation, conclude and give directions of future research.

2. Sufficient Dimension Reduction

In this section we briefly review the Sufficient Dimension Reduction paradigm [9]. Let $\mathbf{x}$ be the set of measurable covariates, with $\mathbf{x} \in \Re^D$, and let $\mathbf{y}$ be the output variables. In supervised learning, the purpose of *sufficient dimension*

Figure 1. **Dynamic textures database**: Illustration of 2 instances of each of the 10 dynamic texture categories, i.e., branches, cloud, lab, flower, fountain, grass, leaves, ripple, sea_anemone, waves. Note that the variation in appearance within a single category is very large.

reduction (SDR) is to estimate a low-dimensional representation $\mathbf{z}$ that is sufficient for the prediction task, with

$$\mathbf{z} = \mathbf{W}\mathbf{x}\,. \tag{1}$$

$\mathbf{W}$ is a projection matrix to a d-dimensional space, and $\mathbf{z} \in \Re^d$, with $d \ll D$. One of the key advantages of SDR with respect to other supervised and unsupervised dimensionality reduction techniques is that it makes no assumption on the form of the distribution of $\mathbf{x}$.

The SDR criterion can be captured formally as the following conditional independence assertion

$$\mathbf{y} \perp\!\!\!\perp \mathbf{x}|\mathbf{z}\,. \tag{2}$$

This means that given $\mathbf{z}$, the remaining features of $\mathbf{x}$ are conditionally independent of the output $\mathbf{y}$. In the statistical sense, $\mathbf{z}$ is sufficient for estimating $\mathbf{y}$.

Kernel Dimension Reduction (KDR) [5] maps the random variables $\mathbf{x}$ and $\mathbf{y}$ to reproducing kernel Hilbert spaces (RKHS) and characterizes conditional independence using cross-covariance operators

$$\Sigma_{yy|x} = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}. \tag{3}$$

Note that $\Sigma_{yy|x} \leq \Sigma_{yy}$ as the second term is positive semidefinite. Intuitively this means that conditioning on $\mathbf{x}$ reduces uncertainty [13].

In [5] it was shown that KDR performs well on a variety of tasks, where the training and testing data are i.i.d. samples from the joint distribution $p(\mathbf{x}, \mathbf{y})$. However, in many computer vision applications, one has to deal with time-series data, where samples are now correlated in time.

3. Sequence Kernel Dimension Reduction

In this section we develop a novel KDR formulation for sequence data. The idea is that we would like the latent coordinates of similar input observations that are close in space, time and/or phase to be close in latent space.

We formulate the *Sequence Kernel Dimension Reduction* (S-KDR) problem of estimating the $\mathbf{W}$ that minimizes $tr[\hat{\Sigma}_{yy|z}]$, where $\hat{\Sigma}_{yy|z}$ is the empirical estimate of $\Sigma_{yy|z}$ as

$$\begin{aligned} \min \quad & tr\left[\mathbf{K}_y^c(\bar{\mathbf{K}}_z^c + \epsilon\mathbf{I})^{-1}\right] \;+\; \lambda R(\mathbf{W}) \\ \text{subject to} \quad & \mathbf{W}^T\mathbf{W} = \mathbf{I} \end{aligned} \tag{4}$$

where $\mathbf{I}$ is the identity matrix, $tr[]$ is the trace, $\Sigma_{yy|z}$ is defined in Eq. (3), and $\mathbf{K}^c$ denotes the centered kernel matrix

$$\mathbf{K}^c = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbf{K}(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)^T \tag{5}$$

with $\mathbf{1}$ a vector of all ones, λ a constant, and R a regularizer.

$\mathbf{K}_y^c$ can be computed using a kernel that measures output label similarities. Here we are interested in multi-class sequence classification. We define a distance metric which is 0 for points of the same class and 1 for points of different classes. Note that this distance metric is equivalent to the Hamming distance between indicator vectors that indicate the class label. To smooth the kernel, we use an RBF kernel on top of this distance metric.

Different strategies can be used to combine the temporal, spatial and phase information. We now propose kernels that capture this information as well as regularizers that exploit the manifold structure of the dynamics.

3.1. Building individual kernels

Probably the simplest way to combine the different sources of information is to build individual kernels and combine them using multiple kernel learning. In particular, we combine them using a product of kernels

$$\bar{k}_z = k_x(\mathbf{x}_i, \mathbf{x}_j) \cdot k_t(t_i, t_j) \cdot k_p(\mathbf{z}_i, \mathbf{z}_j) \tag{6}$$

where $\bar{\mathbf{K}}_z = \{\bar{k}_z(\mathbf{z}_i, t_i, \mathbf{z}_j, t_j)\}$. Note that these kernels are restricted to be Mercer kernels, i.e., the resulting Gram matrix is positive semidefinite for any possible data. We now design suitable kernels for $\mathbf{K}_x$, $\mathbf{K}_t$ and $\mathbf{K}_p$.

Observations: The observation kernel should encourage latent coordinates of similar (input) observations to be similar. We use an RBF kernel such that

$$k_x(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|_2^2}{2\theta_x^2}\right) \tag{7}$$

with $\mathbf{x}_i$ a single frame.

Dynamics: The dynamics kernel should encourage points that are close in time to be close in latent space. We use a bias plus an RBF kernel to model the dynamics

$$k_t(t_i, t_j) = 1 + \exp\left(-\frac{\|t_i - t_j\|_2^2}{2\theta_t^2}\right)\delta_{i,j} \tag{8}$$

where $\delta_{i,j} = 1$ if the i-th and j-th datapoints are from the same sequence, and 0 otherwise.

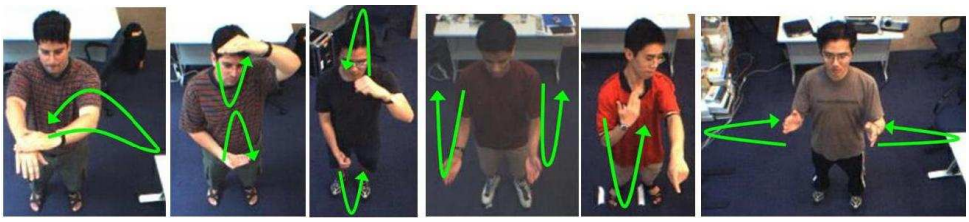


Figure 2. **Arm gesture database**: Illustration of the 6 gesture classes: FB - Flip Back, SV - Shrink Vertically, EV - Expand Vertically, DB - Double Back, PB - Point and Back, EH - Expand Horizontally. Each image is an abbreviation of a gesture class, where the fingertip motion trajectories are depicted in green. The direction of the arrow symbolizes the direction in which the gesture is performed.

Phase: For periodic motions it is desirable for points with similar phase to be close in latent space. If the phase of the observations is known a priori one could build a periodic kernel by mapping the one-dimensional phase variable ϕ into a two-dimensional variable $\mathbf{u}(\phi) = (\cos(\phi), \sin(\phi))$[18] such that

$$\hat{k}_p^c(\mathbf{z}_i, \mathbf{z}_j) = 1 + \exp\left(-\frac{\sin^2(\frac{\phi_i - \phi_j}{2})}{\theta_p^2}\right)\delta_{i,j} \ . \quad (9)$$

Since we only want to align in phase latent coordinates of motions that are from the same sequence, we set $\delta_{i,j} = 1$ when the i-th and j-th datapoints are from the same sequence, and 0 otherwise. While for some applications one can have a reasonable estimate of the phase, in this paper we tackle the more challenging scenario where the phase of the motion is unknown, and has to be estimated at the same time as the embedding. In particular we seek to express the phase of each point as a function of the latent coordinates. Using $\sin^2(\phi) + \cos^2(\phi) = 1$, we can express the kernel in Eq. (9) as a function of the cosine of the phase increment $\phi_{z_1,z_2} = \phi(\mathbf{z}_1) - \phi(\mathbf{z}_2)$. Using the fact that $\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2 \cos(\phi_{ab})$, we can finally write

$$k_p(\mathbf{z}_i, \mathbf{z}_j) = 1 + \exp\left(\frac{1}{2\theta_p^2} \frac{\tilde{\mathbf{z}}_i^T \tilde{\mathbf{z}}_j}{\sqrt{||\tilde{\mathbf{z}}_i||^2 \cdot ||\tilde{\mathbf{z}}_j||^2 + \eta}}\right)\delta_{i,j} \quad (10)$$

where the $\tilde{\mathbf{z}_i}$ are the centered latent coordinates computed as $\tilde{\mathbf{z}}_i = \mathbf{z} - \bar{\mathbf{z}}_{s_i}$, with $\bar{\mathbf{z}}_{s_i}$ the mean value of the latent coordinates of each sequence, and η is a regularization parameter. Note that even though $\mathbf{z} = \mathbf{W}\mathbf{x}$, we have explicitly stated the dependency on the latent space in the kernel. As shown in the experiments, even if this kernel is designed for periodic motions, it can also model non-periodic ones.

3.2. Dynamic Time Warping kernels

The dynamic kernels introduced above encourage points that are close in time, phase and in observation space to be close in latent space. When the dynamics of the different sequences are well structured, one can make use of more sophisticated kernels to capture this structure. Dynamic Time Warping (DTW) solves the problem of computing distances between two sequences of different lengths. This is typically done by solving the following optimization problem

$$\begin{aligned} \min_{\psi,\theta} \quad & \textstyle\sum_{k=1}^{L} d(x_{\psi_k}^{(j)}, x_{\theta_k}^{(p)}) \quad (11) \\ s.t. \quad & 1 \le \psi_1, \psi_L \le |\mathbf{x}^{(j)}|, \psi_i \le \psi_{i+1}, i = 1, ..., L-1 \\ & 1 \le \theta_1, \theta_L \le |\mathbf{x}^{(p)}|, \theta_i \le \theta_{i+1}, i = 1, ..., L-1 \end{aligned}$$

where d is the local distance, typically Euclidean, $\mathbf{x}^{(j)}$ is the j-th sequence, L_j and L_p are the lengths of the two sequences, and $L \le L_j + L_p$ is the number of warping frames. This problem can be solved using dynamic programming.

Once the warpings are estimated, we can compute a DTW kernel by simply converting distances into similarities. We smooth the results of the DTW by convolving the resulting kernel with a Laplace kernel. We can then construct a dynamics kernel by combining the DTW kernel with the kernel defined in the latent space

$$\bar{\mathbf{k}}_z = \mathbf{k}_x(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \cdot [1 \ + \ \mathbf{k}_{DTW}(\mathbf{y}^{(i)}, \mathbf{y}^{(j)})]\delta_{y_i,y_j} \quad (12)$$

with $\delta_{y_i,y_j} = 1$ when $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are from the same class, and 0 otherwise. Note that $\bar{\mathbf{k}}_z$ is a matrix defined for pairs of sequences, unlike the kernel in Eq. (6) that was defined in terms of individual frames. However, the elements of $\bar{\mathbf{k}}_z$, $\bar{\mathbf{k}}_z(\mathbf{x}_r^{(i)}, \mathbf{x}_s^{(j)})$, are still defined on a per frame basis.

3.3. Choice of regularizers

Additional regularizers could be employed in order to exploit the manifold structure underlying the dynamics. In this paper we propose two different regularizations: an L_2 weighted distance and a regularizer based on the Laplacian.

In order to encourage the latent coordinates of warped points to be close in latent space we employ a weighted squared loss, with weights given by the DTW kernel

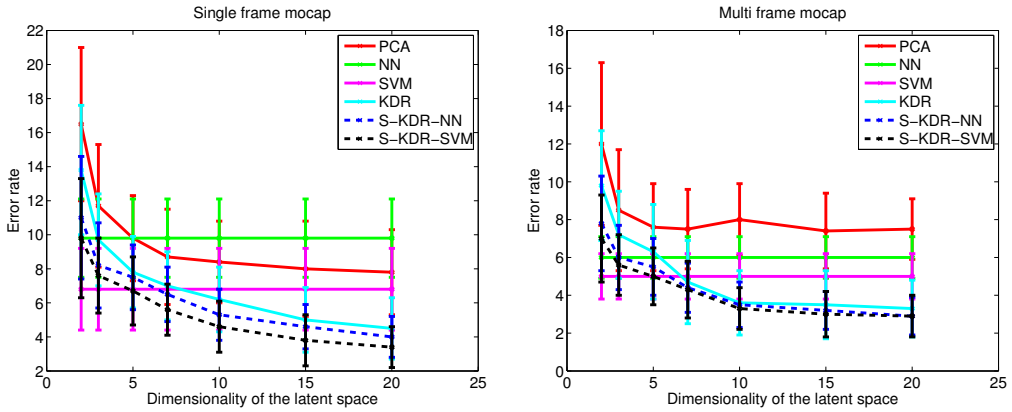
$$R(\mathbf{W}) = \sum_{i,j} (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W} k_{DTW}(\mathbf{x}_i, \mathbf{x}_j) \mathbf{W}^T (\mathbf{x}_i - \mathbf{x}_j)$$

Note that $k_{DTW}(\mathbf{x}_i, \mathbf{x}_j)$ is a scalar.

A widely employed regularizer in semi-supervised learning is the Laplacian. Alternatively we can construct

$$R(\mathbf{W}) = tr\left(\mathbf{W}^T \mathbf{X}^T (\mathbf{D} - \mathbf{K}_{DTW}) \mathbf{X} \mathbf{W}\right)$$

where $\mathbf{D}$ is a diagonal matrix with elements $D_{ii} = \sum_j k_{DTW}(\mathbf{y}_i, \mathbf{y}_j)$. As shown in our experiments, in practice the L_2 regularization outperforms the Laplacian.



(Single frame estimation) (Multi-frame estimation)

Figure 3. **Classification error for the Mocap database** as a function of the latent space dimensionality. The Mocap database consist of walking, running and jumping motions performed by different subjects. The observations are 62D, where the first 3 dimensions represent spatial velocities, and the the remaining degrees of freedom represent joint angles that define the human pose. Our approach performs extremely well with only $3.4 \pm 1.2\%$ classification error for single frame estimation, and $2.9 \pm 1.0\%$ for sequence classification.

3.4. Optimization

As our objective function is nonconvex, we use projected gradient descent with simulated annealing and a line search for minimizing Eq. (4). For all experiments we estimate the parameters using cross-validation. For inference, we use the estimated projection matrix $\mathbf{W}$ to compute the latent coordinates of the test points $\mathbf{x}_*$, such that $\mathbf{z}_* = \mathbf{W}\mathbf{x}_*$, and use NN and SVM as the classifiers in the low dimensional space.

4. Experimental Evaluation

We demonstrate the effectiveness of our approach for classifying motion capture data, categorical dynamic textures and video sequences of human gestures.

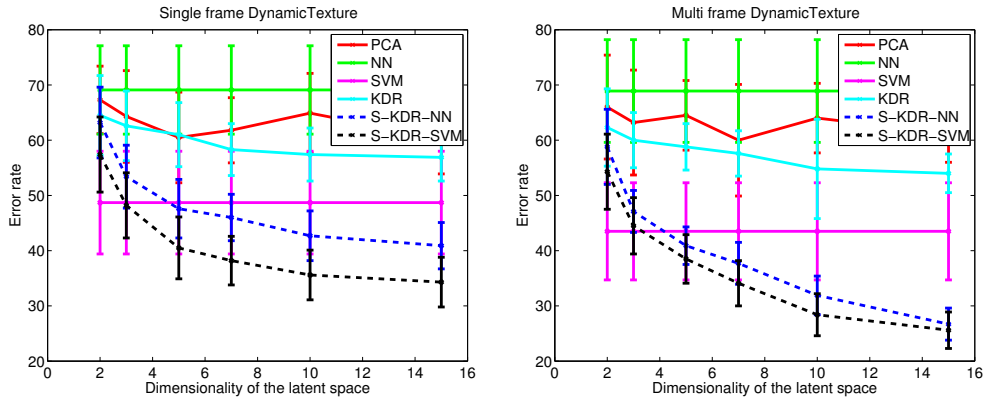
Dynamic Textures: We use the DynTex database [10] to define 10 different categories of dynamic textures: branches, cloud, lab, flower, fountain, grass, leaves, ripple, sea_anemone, waves. We scaled each image to be of dimension 25×25, resulting in 625D observations. For each dynamic texture, we took the first 200 frames subsampled by a factor of 4, so that each sequence has a temporal duration of 50 frames. Fig. 1 depicts examples of the different categories. Note that there is a large intra-class variation. Other results reported on this database are instance level recognition, where each video is segmented and divided into training and testing. In contrast, our experiment is a category recognition experiment.

Arm Gesture Dataset: We use the gesture database of [19] that is composed of six gestures: Expand Horizontally (EH), Expand Vertically (EV), Shrink Vertically (SV), Point and Back (PB), Double Back (DB) and Flip Back (FB). The users were asked to perform these gestures in front of a stereo camera. The stereo-tracking algorithm of [3] was used to estimate the head, torso, arms and forearms. Following [19], for each frame a redundant parameterization composed of joint angles and relative coordinates of the arm joints define the 20D input observations. The gestures were performed by 13 users, and on average 90 gestures were collected per class. Fig. 2 illustrates the different gestures. We subsample the data by a factor of 2; the length of the different gestures varies from 14 to 42 frames.

Head Gesture Dataset: The head gesture data consists of interactions between 16 human participants and an embodied agent [19]. The participants interactions were recorded, resulting in a total of 152 head nods, 11 head shakes and 179 miscellaneous sequences. The gestures were tracked using an adaptive view-based appearance model which captures the user appearance in different poses [12]. The observations consist of the FFT of the 3D angular velocities recovered by the tracker. Each observation forms a 51D vector. We subsample the data by a factor of 2; the length of the gestures varies dramatically from 18 to 908 frames.

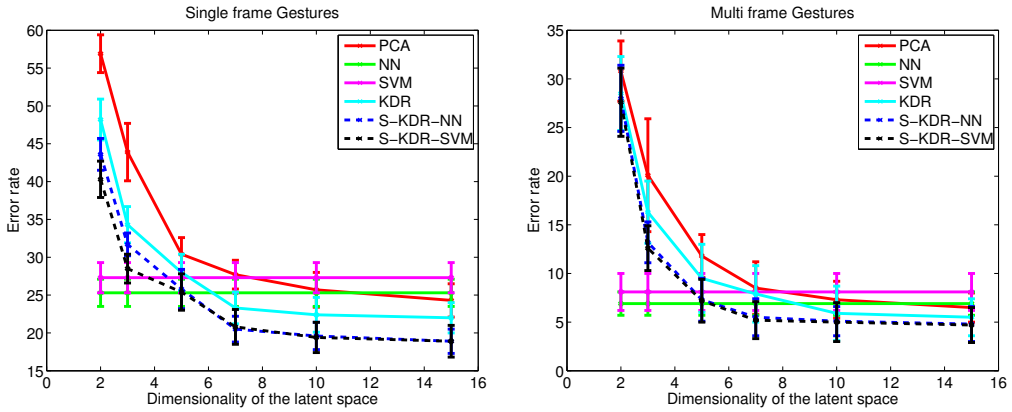
Mocap data: We use motion capture data of walking, running and jumping performed by different subjects from the CMU mocap database [11]. Each observation is a 62D vector, where the first 3 dimensions are the spatial velocities, and the remaining dimensions are joint angles that characterize the pose. We subsample the mocap data by a factor of 4 so that the framerate is 30Hz. The length of the different sequences varies from 65 to 100 frames.

Activity Recognition: The activity recognition benchmark of [6] consists of 9 different subjects performing 10 different actions, including running, walking, skipping, jumping jack, waving and bending. The video sequences are low-resolution (180×144) with relatively uniform background and stationary camera. We first perform segmentation by utilizing background substraction on the joint color and motion (i.e., optical flow) space. Based on the segmented video sequences, we normalize the bounding boxes



(Single frame estimation) (Multi-frame estimation)

Figure 4. **Classification error for the Dynamic texture dataset** of [10] as a function of the latent space dimensionality. The DynTex dataset consists of video sequences of natural scenes, exhibiting both periodic and non-periodic motions. We labeled the dataset with 10 categories, each with a variety of textures. Due to the complexity and high dimensionality of the database, the classification task is inherently difficult and all the baselines have errors well larger than 50%. Our approach is able to capture the underlying dynamics and achieve $34.3 \pm 4.5\%$ classification error for single frame estimation, and $25.6 \pm 3.3\%$ for sequence classification.



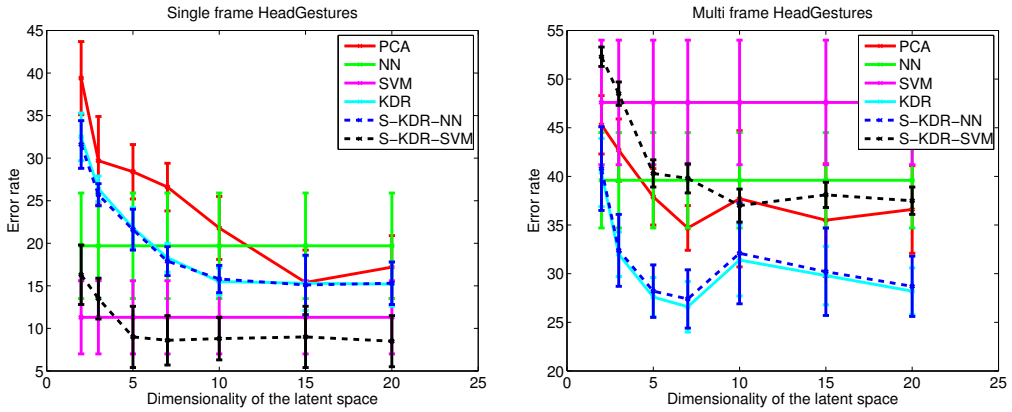
(Single frame estimation) (Multi-frame estimation)

Figure 5. **Classification error for the Arm Gestures dataset** of [19] as a function of the latent dimensionality. The dataset consists of 6 different hand motions performed by different subjects. Our approach results in good performance with $18.9 \pm 1.6\%$ classification error for single frame estimation and $4.7 \pm 1.8\%$ for sequence classification.

and compute HOG features by overlaying 7×9 cell blocks with 5 histogram channels. The resulting feature space is of dimension 315. Finally, we truncate the sequences at 50 frames and subsample them by a factor of 2, as this is enough to capture the dynamics.

For all databases, we compare our approach (i.e., multiple kernel learning of Eq. 6) to the following baselines: classification in the observation space using NN and non-linear SVMs, PCA, SVM-HMM [1], and the original KDR [5]. SVM-HMM discriminatively trains a k-th order Hidden Markov Model (HMM) using the Structural Support Vector Machine formulation (SVM-Struct) [16]. Given an input sequence of feature vectors, the model predicts a sequence of labels according to a linear discriminant function. SVM-HMM learns an emission weight vector for each k-th order label sequence and one transition weight vector between adjacent labels. We report results of classifying each data point independently, and combining the classifiers from the whole sequence by voting. Note that the latter assumes that the test data is segmented. The error bars in all figures represent ± 1 standard deviations. To avoid clutter, the SVM-HMM baseline is only shown in the text. The performance of SVM-HMM is consistently worse than SVM in the observation space.

Fig. 3 depicts classification error averaged over 5 splits as a function of the latent space dimensionality for the mocap database. For each class, 5 examples were used for training and 20 for testing. Our approach consistently outperforms all the baselines even when using very low-dimensional spaces. Note that NN and SVM in the observation space and SVM-HMM accuracies do not vary with the dimensionality since these methods do not learn a latent space. The average error of SVM-HMM was $7.4 \pm 1.2\%$ for single frame and $5.2 \pm 1.2\%$ for multi-frame. Fig. 3 (left) depicts the error rate when doing single frame classification, i.e., every frame is independent. Fig. 3 (right) shows



(Single frame estimation) (Multi-frame estimation)

Figure 6. **Classification error for the Head Gestures dataset** of [19]. The dataset consists of 3 classes of head motions - shakes, nods and other movements. The motions have considerable overlap, and sequence classification is intrinsically more difficult than frame estimation. Our approach achieves $8.5 \pm 3.0\%$ classification error for single frame and $28.7 \pm 3.1\%$ for sequence estimation.

	N	Accuracy (%)
HMM $w = 0$	80	84.22
CRF $w = 0$	80	86.03
CRF $w = 1$	80	81.75
HCRF (one-vs-all) $w = 0$	80	87.49
HCRF (multi-class) $w = 0$	80	91.64
HCRF (multi-class) $w = 1$	80	93.85
S-KDR-SVM $w = 0$	10	**95.3**

Table 1. Comparison of our approach (S-KDR)) HMM, CRF and HCRF. We include the classification accuracy reported in [19]. Our approach outperforms all the baselines, and only requires 10 training examples per class instead of 80 that were used to train the baselines. $w = 1$ means that the previous and next observations are concatenated to produce an observation.

	N	Accuracy (%)
HMM $w = 0$	171	65.33
CRF $w = 0$	171	66.53
CRF $w = 1$	171	68.24
HCRF (multi-class) $w = 0$	171	71.88
HCRF (multi-class) $w = 1$	171	85.25
S-KDR-SVM $w = 0$	15	**91.5**

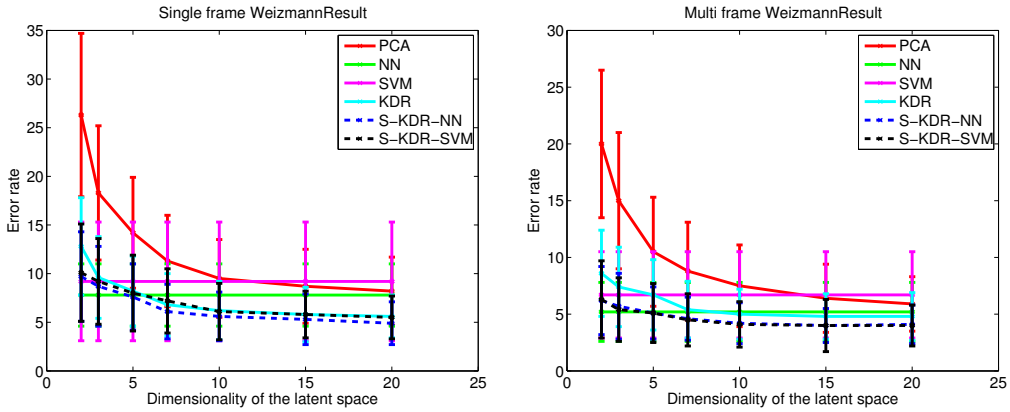
Table 2. Comparison of our approach (S-KDR) with HMM, CRF and HCRF. We include the classification accuracy reported in [19]. N denotes the total number of training examples.

classification error when combining the single frame classifiers by voting. As expected voting results in better performance since it combines information from all the frames in the sequence. However, the single frame estimation does not assume that the sequences are segmented. Our approach results in extremely good performance; $3.4 \pm 1.2\%$ classification error for single frame estimation, and $2.9 \pm 1.0\%$ for sequence classification.

Fig. 4 shows classification error averaged over 5 splits as a function of the dimensionality of the latent space for the dynamic texture database [10]. 3 examples per class were used for training and 5 for testing. The correct classification rate of SVM-HMM is $36.1 \pm 5\%$ for single frame and $34.4 \pm 7.2\%$. Our approach significantly outperforms all the baselines even with low-dimensional latent spaces. Note that this is an extremely hard problem since one has to classify 10 dynamic texture categories with very few examples and very large intra-class variations, as shown in Fig. 1. As a result, the performance of the baselines is as low as 34%, while for our approach is 75%.

Fig. 5 shows classification error for the arm gestures dataset of [19] averaged over 5 splits. For each class 10 examples were used for training and 100 for testing. The different gestures are shown in Fig. 2. The performance of SVM-HMM is $67.5 \pm 3.2\%$ for single frame and $82.2 \pm 3.5\%$ for multi frame. Our approach outperforms the baselines when using single frame or multi-frame (voting) classification, resulting in $81.1 \pm 1.6\%$ for single frame and $95.3 \pm 1.8\%$ for multi-frame. As shown in Table 1, our approach also results in better performance than HMMs, CRFs and HCRFs [19]. Moreover, we only require 10 training examples per class, while the baselines were trained with approximately 80 examples per class. Note also that there is a large benefit for this database when using information from multiple frames. This is because, even though the different gestures have some poses in common, the overall gesture is very discriminative.

Error rates averaged over 5 splits for the head gesture database of [19] are shown in Fig. 4. For each class 5 examples are used for training and 30 for testing. Note that, unlike with the other databases, incorporating information from multiple frames decreases performance. This is to be expected since head nods, head shakes and miscellaneous have very similar poses, sometimes for more than 50% of the length of the sequence. Moreover, failures in the monocular tracking make SVM approaches fail in the multi-frame setting. The mean error of the SVM-HMM is $28.8 \pm 5.7\%$ for single frame and $39.1 \pm 4.3\%$ for multi-frame. Comparisons of our approach to HMMs, CRFs and HCRFs are



(Single frame estimation) (Multi-frame estimation)

Figure 7. **Classification error for the Weizmann dataset** of [6]. The dataset consists of 10 different actions. Our approach achieves $4.9 \pm 2.2\%$ classification error for single frame and $4.0 \pm 1.8\%$ for sequence estimation.

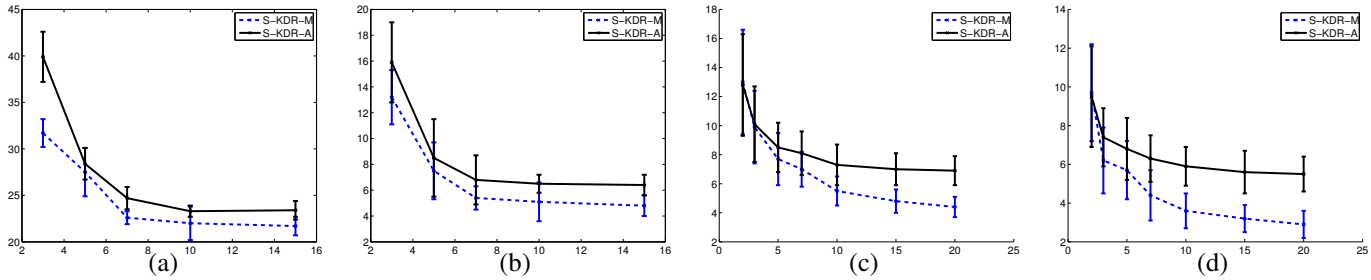


Figure 8. **Multiplicative vs additive kernels**. We compare the results obtained with the kernel of Eq. (6) against using an additive kernel. Classification errors for (a) single frame arm gesture data (b) multi-frame arm gesture data (c) single frame mocap data and (d) multi-frame mocap data are depicted. Our multiplicative kernel outperforms the additive one in all tasks.

shown in Table 2. Our approach outperforms all the baselines and it is trained using only 10% of the data used to train the other baselines, resulting in $8.5 \pm 3\%$ mean error for single frame and $28.7 \pm 3.1\%$ for multi-frame.

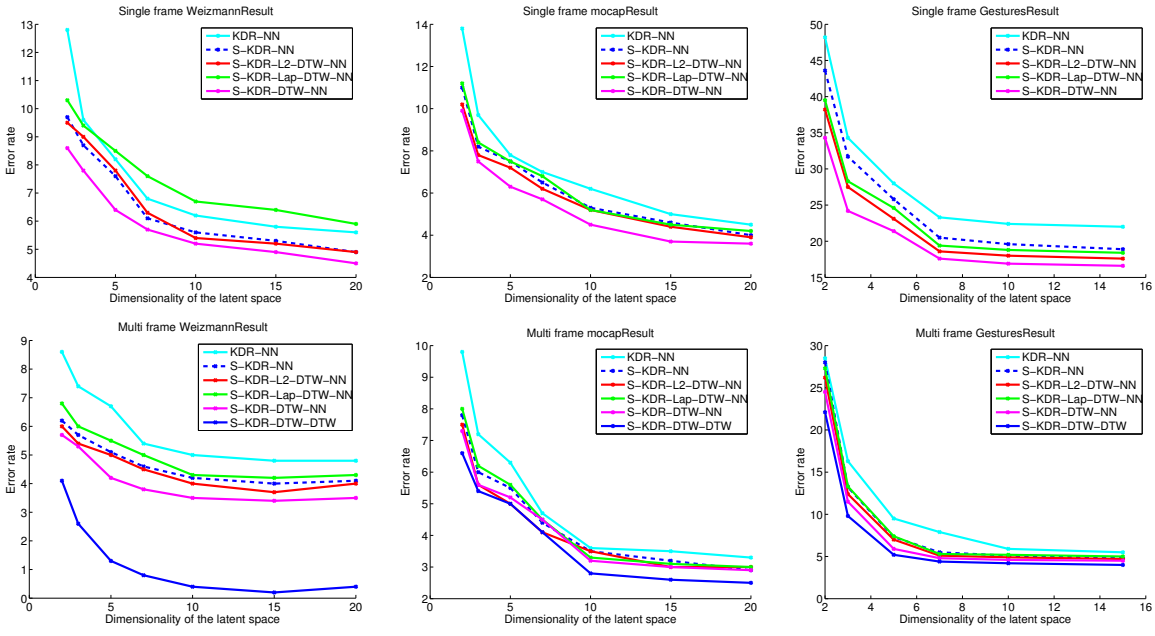
Fig. 4 shows classification error averaged over 5 splits for the Weizmann dataset. For each class, 4 sequences were used for training and 5 for testing. The average error rate of SVM-HMM is $9.8 \pm 5.8\%$ and $7.0 \pm 3.8\%$ for single-frame and multi-frame estimation, respectively. Note that SVM in the original space overfits, and NN works better. Our approach consistently outperforms PCA and SVM, with a $4.9 \pm 2.2\%$ error rate for single frame estimation and a $4.1 \pm 1.7\%$ error rate for sequence classification.

Not only does our approach outperform the baselines, but, more importantly, the standard errors are much smaller. This implies that S-KDR consistently learns latent spaces that are good for the classification task. We also investigate other ways of combining the different sources of information. In particular, we compare the multiplicative kernel of Eq. (6) to an additive kernel, $\bar{\mathbf{K}}_z = \mathbf{K}_x + \mathbf{K}_t + \mathbf{K}_p$. As shown in Fig. 8 the multiplicative kernel outperforms the additive one.

We now evaluate the effectiveness of the dynamic time warping kernels. For the Weizmann, Mocap and the Arm Gesture datasets, the dynamics are well-structured and relatively distinct, allowing for accurate computation of the time warpings. For each of these datasets, we compare KDR and S-KDR to 3 different ways of incorporating DTW: L_2 regularization (S-KDR-L2-DTW), Laplacian regularization (S-KDR-Lap-DTW) and the kernel of Eq. (12) (S-KDR-DTW). In the former two cases we add the regularizations to the KDR objective, with $\bar{\mathbf{k}}_z = \mathbf{k}_x$. We believe that this is a fair comparison with S-KDR, since then the dynamic information is utilized only once. In all three cases, we find that NN outperforms SVM; as a consequence we only report NN results. Furthermore, since the new latent spaces are optimal for sequence alignment, we perform an additional per-sequence classification using DTW in the latent space. As shown in Fig. 4 the results for all three datasets are similar: L_2 regularization typically improves the S-KDR performance while the Laplacian regularization often degrades it. The kernel combination in Eq. (12) uniformly achieves the best performance. Moreover, the DTW classifier in the latent space achieves state-of-the-art results on the Weizmann dataset (i.e., 99.8%). This suggests that when there is sufficient structure in the dynamics, the DTW kernel correctly captures both linear and nonlinear aspects of dynamics, and DTW is the optimal classifier.

5. Conclusion

In this paper we have developed a novel Kernel Dimension Reduction formulation for time series data. Our approach combines spatial, temporal and periodic information



(Weizmannn) (Mocap) (Arm Gesture)

Figure 9. **Classification error for the Weizmann [6], Mocap [11] and Arm Gesture [19] datasets.** Comparison of different methods of incorporating dynamics structure with the DTW kernel. Using the DTW kernel consistently achieves the best performance.

in a principled manner, and learns an optimal embedding for the end-task. We have demonstrated the effectiveness of our approach in classifying motion capture data, categorical dynamic textures, human gestures and activities from video. Our approach outperforms a large variety of baselines comprising unsupervised learning (i.e., PCA), classification in the observation space and the SVM-HMM. When compared to sequence classification methods, i.e., HMMs, CRFs and HCRFs, our approach performs similarly or better while requiring much smaller training sets. We are planning to explore sparsification techniques [2] for fast learning and the development of new kernels for combining other sources of information.

References

[1] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *ICML*, 2003.

[2] J. Candela and C. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1935–1959, 2005.

[3] D. Demirdjian and T. Darrell. 3-D articulated pose tracking for untethered deictic reference. In *ICMI*, 2002.

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

[5] K. Fukumizu, F. Bach, and M. Jordan. Kernel dimension reduction in regression. *Annals of Statistics*, 37:1871–1905, 2009.

[6] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *PAMI*, 29:2247–2253, 2007.

[7] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[8] N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[9] K.-C. Li. Sliced inverse regression for dimension reduction. *J. Amer. Statist. Assoc.*, 86:316–327, 1991.

[10] DynTex. http://old-www.cwi.nl/projects/dyntex/database.html.

[11] Mocap. http://mocap.cs.cmu.edu.

[12] L. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance model. In *CVPR*, 2003.

[13] J. Nilsson, F. Sha, and M. Jordan. Regression on manifolds using kernel dimension reduction. In *ICML*, 2007.

[14] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[15] J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[16] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[17] R. Urtasun and T. Darrell. Discriminative Gaussian process latent variable models for classification. In *ICML*, 2007.

[18] R. Urtasun, D. Fleet, A. Geiger, J. Popovic, T. Darrell, and N. D. Lawrence. Topologically-constrained latent variable models. In *ICML*, 2008.

[19] S. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *CVPR*, 2006.