Kernel-Based Sparse Representation for Gesture Recognition

Yin Zhou[a], Kai Liu[a,b,*], Rafael E. Carrillo[a], Kenneth E. Barner[a,*], Fouad Kiamilev[a]

[a]*Department of ECE, University of Delaware, Newark, DE, 19716, USA*
[b]*School of Electrical Engineering and Information, Sichuan University, 610065, China*

Abstract

In this paper, we propose a novel sparse representation based framework for classifying complicated human gestures captured as multi-variate time series (MTS). The novel feature extraction strategy, CovSVDK, can overcome the problem of inconsistent lengths among MTS data and is robust to the large variability within human gestures. Compared with PCA and LDA, the CovSVDK features are more effective in preserving discriminative information and are more efficient to compute over large-scale MTS datasets. In addition, we propose a new approach to kernelize sparse representation. Through kernelization, realized dictionary atoms are more separable for sparse coding algorithms and nonlinear relationships among data are conveniently transformed into linear relationships in the kernel space, which leads to more effective classification. Finally, the superiority of the proposed framework is demonstrated through extensive experiments.

Keywords: Gesture recognition, Computer vision, Compressive sensing, Sparse

*Corresponding author. Tel: 1-302-8312405

Email addresses: zhouyin@udel.edu (Yin Zhou), kailiu@scu.edu.cn (Kai Liu), carrillo@ee.udel.edu (Rafael E. Carrillo), barner@udel.edu (Kenneth E. Barner), kiamilev@udel.edu (Fouad Kiamilev)

representation, Dictionary learning

1. Introduction

Sparse representation has achieved state-of-the-art results in many fields, such as image compression and denoising [1], face recognition [2, 3], video-based action classification [4], etc. The success of this technique is partially due to its robustness to noise and missing data. For example, sparse representation-based classification (SRC) [2] yields impressive results in face recognition by encoding a query face image over the entire set of training template images and identifying the label of the query sample by evaluating which class yields the minimum reconstruction error. However, little effort has been made to apply this technique to classifying multi-variate time series (MTS) data.

Classifying multivariate time series (MTS) is a challenging task in many areas, e.g., pattern recognition [5] and computer vision [6]. An MTS is an $m \times n$ matrix, where m is the number of observations on an individual event captured by sensors such as video cameras, position trackers and cybergloves, while n denotes the number of independent attributes [7], also known as variables [5, 8] or features [9, 10]. For each MTS, m is typically varying due to different motion durations for each instance, while the number of attributes, n, is the same for all the series since they are recorded by the same set of devices. For conventional feature extraction methods, e.g., PCA and LDA, downsampling and interpolation are usually applied on each MTS in order to normalize the data length. However, downsampling may cause a loss of salient information [5], while interpolation may induce distortion to the original data [8].

Gesture MTS data possess both spatial and temporal information. While spa-

tial information depicts the entire static pattern, temporal information contains the dynamic dependencies between adjacent recordings. Algorithms that exploit chronological order within time series, e.g., Dynamic Time Warping (DTW) [11, 12] and Longest Common Subsequence (LCSS) [13], assume that similar signals must be recorded in the same order. However, motion order and direction may vary significantly among users presenting the same gesture. Consequently, such algorithms need to store all possible permutations of each gesture in memory and conduct pair-wise matching during recognition, resulting in excessive computation and storage requirements [14]. For example, a 2-stroke letter "t" requires $2! \times 2^2 = 8$ permutations to represent all possibilities, while an l-stroke gesture takes $l! \times 2^l$ permutations.

Notably, real-world gestures and movements, such as human gait and sign language, are performed according to a strict "grammar". This observation indicates that effectively distinguishing complicated spatial patterns is the key to successful recognition, rather than exploiting temporal order [7, 5, 8]. Motivated by this observation and reasoning, we consider feature extraction for MTS data ignoring the temporal ordering. More specifically, we generalize the capability of SRC to classifying MTS data.

The performance of SRC relies on the quality of the dictionary. We propose a novel feature extraction technique, called Covariance Matrix Singular Value Decomposition for Kernelization (CovSVDK), which possesses three notable merits: CovSVDK is 1) invariant to inconsistent lengths and temporal disorder across MTS data; 2) robust to the large variability within human gestures; 3) efficient to compute. In particular, the robustness of the feature extraction strategy is attributed to the fact that CovSVDK essentially enforces ℓ_1 minimization algorithms

to favor training samples that are consistently close to the query sample in every sub-feature space. Moreover, we propose a new approach to kernelize sparse representation. With this method, dictionary atoms are more separable for sparse coding algorithms and nonlinear relationships among data can be conveniently transformed into linear relations in kernel space, which leads to more effective classification. Finally, we evaluate the proposed framework over extensive datasets. For the Georgia-Tech HG database, a 100% recognition rate is stably achieved; over the High-quality Australian Sign Language (HAuslan) database, the recognition accuracy is greater than 91.2%; for the univariate UCR Time-Series Repository, the proposed classifier outperforms competing methods by achieving the lowest error rate on 10 out of 20 datasets.

The remainder of the paper is organized as follows. First, we give a brief review of related work and establish the problem formulation in Section 2. In Section 3, the proposed method is presented. Experiments and comparison with existing methods are presented in Sections 4 and 5. Finally, we summarize in Section 6 and note future directions.

2. Related Work and Problem Formulation

2.1. Related Work

Many algorithms have been proposed to measure the similarity among multi-dimensional time series, e.g., Hidden Markov models (HMMs) [15], DTW [11, 12], LCSS [13], and Mixture of Bayes Network Classifier [9], among others. Principal components (PCs) based methods are, perhaps, the most widely known similarity measure for multi-attribute time series, with the approach first defined by Krzanowski [16] in 1979. Many subsequent PC efforts focused on comput-

ing the similarity value using different weighting strategies to aggregate the inner products between PC pairs [5, 7, 8].

For instance, Li *et al.* proposed a similarity measure for motion streams using only the largest singular value and the corresponding singular vector [7]. In [6], the authors further proposed k Weighted Angular Similarity (kWAS) by considering the k largest singular value/vector pairs. Yang and Shahabi [5] proposed a similarity measure, called Extended Frobenius norm (Eros), which included all the singular values by employing a heuristic aggregating function to compute universal weights for all MTS data. The similarity measure is a weighted sum of inner products between each pair of singular vectors. In practice, however, variance is highly concentrated in the several largest eigenvalues and the small values are typically considered as redundancy or noise. Hence, Eros is vulnerable to noise. Yang and Shahabi further further extended their approach by using Eros for Kernel PCA, termed KEros [8].

Recently, some researchers reported the limitation of SRC [2] in classifying nonlinear data. Zhang *et al.* [17] proposed the kernel sparse representation-based classifier (KSRC) by introducing the kernel trick. However, their approach relies on kernel-based dimensionality reduction techniques and thus does not offer a direct generalization to sparse representation in kernel space. Gao *et al.* [18] proposed kernel sparse representation (KSR). However, the KSR objective function cannot be solved by standard sparse coding algorithms as it requires solving a quadratic programming (QP) problem, which is of higher computational complexity than ℓ_1 minimization.

2.2. Problem Formulation

In a k-label MTS data classification problem, we define the training set as $\mathbf{T} = \bigcup_{i=1}^{k} \mathbf{T}_i$, where $\mathbf{T}_i = \bigcup_{j=1}^{n_i} \mathbf{t}_{i,j}$ is a subset for the i-th class with n_i samples, and define the query sample as $\mathbf{x}$. Also, denote $N = \sum_i^k n_i$ as the total number of training samples.

There is significant current interest in using SRC [2] to classify audio, image and video signals. It is therefore desirable to explore its capability in the field of MTS data classification. To achieve this goal, several important issues must be addressed: 1) An effective feature extraction method is needed to process large-scale MTS datasets. The method should be efficient in computation and memory consumption, and invariant to inconsistent lengths and temporal disorder across MTS samples. 2) A general formulation of sparse representation suitable for various pattern recognition tasks is also desired. SRC assumes that training atoms reside on a linear manifold and are distinguishable by ℓ_1 minimization algorithms. While this premise holds for face images, it does not necessarily hold for other types of data.

3. Proposed method

This section details methods for effectively extracting MTS data features and present a novel approach to kernelizing sparse representation for classification.

3.1. Feature Extraction for MTS Data

3.1.1. SVD Properties of MTS Data

For an $m \times n$ MTS $\mathbf{t}$ with m observations and n attributes, m is typically much larger than n and varies across different samples. In order to avoid performing SVD on m-varying $\mathbf{t}$, we treat each attribute (columns in the $\mathbf{t}$) as a random

variable and compute the covariance matrix of $\mathbf{t}$ as

$$\mathbf{\Sigma}_t = \mathbf{E}[\mathbf{t}^T\mathbf{t}] - \mathbf{E}^T[\mathbf{t}]\mathbf{E}[\mathbf{t}], \tag{1}$$

where $\mathbf{E}[\cdot]$ denotes the mathematical expectation and $\mathbf{\Sigma}_t$ is of fixed dimension $n \times n$ (here $n \geq 2$). By calculating the $\mathbf{\Sigma}_t$ of $\mathbf{t}$, we discard the ordering information and thus overcome the problem of temporal disorder across MTS samples, since each entry in $\mathbf{\Sigma}_t$ is an inner product between two columns in $\mathbf{t}$ that is invariant to the row-switching of $\mathbf{t}$.

Applying SVD to the covariance matrix yields $\mathbf{\Sigma}_t = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ is a singular vector matrix with orthonormal columns and $\mathbf{\Lambda} = \texttt{diag}(\rho)$ with $\rho = [\lambda_1, \ldots, \lambda_n]^T$ being a vector with singular values descendingly sorted. `diag` is the operator that transforms ρ into a diagonal matrix by putting entries of ρ along the main diagonal in the matrix. Similarly, the covariance matrix $\mathbf{\Sigma}_p$ of MTS $\mathbf{p}$ can be expressed as $\mathbf{\Sigma}_p = \mathbf{V}\mathbf{\Omega}\mathbf{V}^T$, where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ and $\mathbf{\Omega} = \texttt{diag}(\eta)$ with $\eta = [\omega_1, \ldots, \omega_n]^T$. Since $\mathbf{\Sigma}$ is positive semi-definite, its SVD is equivalent to eigenvalue decomposition.

If two MTS $\mathbf{t}$ and $\mathbf{p}$ are similar to each other, $\|\mathbf{\Sigma}_t - \mathbf{\Sigma}_p\|_F$ should be close to zero. In other words, the singular vector $\mathbf{u}_i$ of $\mathbf{\Sigma}_t$ should resemble $\mathbf{v}_i$ of $\mathbf{\Sigma}_p$ in direction and the singular value λ_i of $\mathbf{\Sigma}_t$ should also be close to ω_i of $\mathbf{\Sigma}_p$. Further discussions on the SVD properties of MTS can be found in Appendix A.

3.1.2. Simple features for sparse representation

For simplicity, we indicate the i-th training sample as $\mathbf{t}_i$. Applying SVD to the covariance matrix, we get $\mathbf{\Sigma}_{t_i} = \mathbf{U}_i\mathbf{\Lambda}_i\mathbf{U}_i^T$, where $\mathbf{U}_i = [\mathbf{u}_i^1, \ldots, \mathbf{u}_i^n]$ and $\mathbf{\Lambda}_i = \texttt{diag}(\rho_i)$ with $\rho_i = [\lambda_i^1, \ldots, \lambda_i^n]^T$. Note that $\mathbf{u}_i^j \in \mathbb{R}^n$ and λ_i^j stand for the j-th

singular vector (principle component) and the j-th singular value of $\mathbf{t}_i$ respectively. We denote $\mathbf{B}^j = [\mathbf{u}_1^j, \mathbf{u}_2^j, \ldots, \mathbf{u}_N^j] \in \mathbb{R}^{n \times N}$ as the dictionary containing the j-th singular vectors extracted from all $\mathbf{t}_i$ with $\|\mathbf{u}_i^j\|_2 = 1$, for $i = 1, \ldots, N$.

Given a query sample $\mathbf{x}$ and corresponding $\mathbf{\Sigma}_x = \mathbf{V}\mathbf{\Omega}\mathbf{V}^T$, denote the j-th singular vector of $\mathbf{x}$ as $\mathbf{v}^j$ and let $\eta = [\omega_1, \ldots, \omega_n]^T$ be the vector containing all the singular values in Ω sorted in the descending order. A simple strategy for classifying $\mathbf{x}$ is to treat a particular $\mathbf{v}^j$ as the feature of $\mathbf{x}$ and employ SRC [2] to identify the feature by solving

$$\alpha^j = \arg\min_{\alpha^j} \|\alpha^j\|_1 \quad \text{subject to} \quad \mathbf{B}^j\alpha^j = \mathbf{v}^j, \tag{2}$$

Obtaining $\alpha^j \in \mathbb{R}^N$, $\mathbf{x}$ can be classified by evaluating the class-wise reconstruction error based on $\mathbf{B}^j$.

The above strategy using one singular vector (e.g., the top one) may work properly with well-separated data. However, real-world gesture recordings are always vulnerable to noise or large variability among individuals. Therefore it is desirable to take into account several most important singular vectors to improve the robustness of the algorithm. In addition, the discriminative information within the singular values should also be exploited.

3.1.3. Robust features for sparse representation

Consider a robust feature vector constructed by unifying the top s singular values and the associated singular vectors ($s \leq n$). Suppose that we have obtained α^j by solving Eq. (2), for all $j = 1, \ldots, s$. Without violating the equality in the

constraint of Eq. (2), we can equivalently rewrite $\mathbf{B}^j\alpha^j = \mathbf{v}^j$ as

$$\hat{\mathbf{B}}^{\mathbf{j}}\hat{\alpha}^j = [\frac{\lambda_1^j}{\|\rho_1\|_2}\mathbf{u}_1^j, \frac{\lambda_2^j}{\|\rho_2\|_2}\mathbf{u}_2^j, \ldots, \frac{\lambda_N^j}{\|\rho_N\|_2}\mathbf{u}_N^j]\hat{\alpha}^j = \frac{\omega^j}{\|\eta\|_2}\mathbf{v}^j \tag{3}$$

where $\hat{\alpha}^j = \Delta\alpha^j$ with $\Delta = \texttt{diag}([\frac{\omega^j\|\rho_1\|_2}{\lambda_1^j\|\eta\|_2}, \ldots, \frac{\omega^j\|\rho_N\|_2}{\lambda_N^j\|\eta\|_2}])$. Applying the same procedure to each pair of $\mathbf{B}^j$ and $\mathbf{v}^j$ for all $j = 1, \ldots, s$, we get

$$\begin{aligned}
\hat{\mathbf{B}}^1\hat{\alpha}^1 &= \frac{\omega^1}{\|\eta\|_2}\mathbf{v}^1 \\
\hat{\mathbf{B}}^2\hat{\alpha}^2 &= \frac{\omega^2}{\|\eta\|_2}\mathbf{v}^2 \\
\ldots &= \ldots \\
\hat{\mathbf{B}}^s\hat{\alpha}^s &= \frac{\omega^s}{\|\eta\|_2}\mathbf{v}^s
\end{aligned} \tag{4}$$

Ideally, if $\mathbf{x}$ is sufficiently similar to $\mathbf{t}_i$, $\mathbf{v}^j$ should resemble $\mathbf{u}_i^j$, so should ω^j and λ_i^j for all $j = 1, \ldots, s$. Therefore, in reconstructing each $\mathbf{v}^j$, the $\mathbf{u}_i^j$ of $\mathbf{t}_i$ should be coded with large coefficient. In other words, if each $\mathbf{u}_i^j$ of $\mathbf{t}_i$ contributes most in representing $\mathbf{v}^j$ of $\mathbf{x}$, $\mathbf{t}_i$ should be similar to $\mathbf{x}$. Then, the class to which $\mathbf{t}_i$ belongs should yield the minimum error in reconstructing $\mathbf{x}$, which indicates that $\mathbf{x}$ is of the same label as $\mathbf{t}_i$.

Motivated by this intuition, we enforce each $\mathbf{v}^j$ of $\mathbf{x}$ to be represented via a universal sparse code α over the corresponding $\hat{\mathbf{B}}^j$. By substituting $\hat{\alpha}^j$ with α for all $j = 1, \ldots, s$, Eq. (4) can thus be simplified as

$$[\hat{\mathbf{B}}^{1^T}, \hat{\mathbf{B}}^{2^T}, \ldots, \hat{\mathbf{B}}^{s^T}]^T\alpha = [\frac{\omega^1}{\|\eta\|_2}\mathbf{v}^{1^T}, \frac{\omega^2}{\|\eta\|_2}\mathbf{v}^{2^T}, \ldots, \frac{\omega^s}{\|\eta\|_2}\mathbf{v}^{s^T}]^T, \tag{5}$$

where $[\hat{\mathbf{B}}^{1^T}, \hat{\mathbf{B}}^{2^T}, \ldots, \hat{\mathbf{B}}^{s^T}]^T$ is a vertical concatenation of all the sub-matrices

$\hat{\mathbf{B}}^j$ and the right hand side is a super-vector by concatenating all $\mathbf{v}^j$. Thus the classification scheme based on unifying the top s pairs of singular values/vectors can be formulated as

$$\alpha = \arg\min_{\alpha} \|\alpha\|_1 \quad \text{subject to} \quad \text{Eq. (5)}, \tag{6}$$

where columns in $[\hat{\mathbf{B}}^{1^T}, \hat{\mathbf{B}}^{2^T}, \ldots, \hat{\mathbf{B}}^{s^T}]^T$ are normalized to unit ℓ_2-norm.

Definition 1 (CovSVDK). *Given an MTS* $\mathbf{t}$*, its covariance matrix is decomposed as* $\mathbf{\Sigma}_t = \boldsymbol{U}\mathbf{\Lambda}\boldsymbol{U}^T$ *by SVD, where* $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$ *is a singular vector matrix with orthonormal columns and* $\mathbf{\Lambda} = \texttt{diag}(\rho)$ *with* $\rho = [\lambda_1, \ldots, \lambda_n]^T$ *is a diagonal matrix with singular values descendingly sorted on the main diagonal. The CovSVDK feature for* $\boldsymbol{t}$ *is defined as*

$$\phi(\boldsymbol{t}) = \left[\frac{\lambda_1}{\| \rho \|_2}\boldsymbol{u}_1^T, \frac{\lambda_2}{\| \rho \|_2}\boldsymbol{u}_2^T, \ldots, \frac{\lambda_s}{\| \rho \|_2}\boldsymbol{u}_s^T\right]^T \in \mathbb{R}^{sn}, \tag{7}$$

where s *subjects to*

$$s = \arg\min \left\{ \frac{\sum_{i=1}^{s} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \geq c \right\} \tag{8}$$

for a pre-selected energy threshold, c*.*

In practice, it is common to empirically set a universal s for all MTS data such that most energy is preserved within the top s singular values. The name CovSVDK stands for Covariance Matrix SVD for Kernelization.

Definition 2. *Given* s*, define* Φ *as a collection of features extracted from the training set* $\boldsymbol{T}$ *according to Definition 1, and write* Φ *as*

$$\Phi = [\phi(\boldsymbol{t}_{1,1}), \ldots, \phi(\boldsymbol{t}_{i,1}), \ldots, \phi(\boldsymbol{t}_{i,n_i}), \ldots, \phi(\boldsymbol{t}_{k,n_k})] \in \mathbb{R}^{sn \times N}. \tag{9}$$

Furthermore, define $\mathbf{y} = \phi(\mathbf{x})$ *as the feature of the query sample* **x.**

Discussion: If we define $r = \max(mn, N)$ and denote d as the reduced dimension, PCA is of computational complexity $O(r^2 d)$ while CovSVDK is of complexity $O(n^2 dN)$. For the cases where m or N is large, $O(r^2 d) \gg O(n^2 dN)$. Thus, CovSVDK is substantially more efficient than PCA over large-scale datasets or for MTS data with long durations. More importantly, the memory usage by PCA is proportional to N^2 or $m^2 n^2$ while the memory consumption by CovSVDK is proportional to n^2. Hence, CovSVDK is also more memory efficient than PCA.

Revisiting Eq. (5), we can substitute $\mathbf{y}$ for $\left[\frac{\omega^1}{\|\eta\|_2}\mathbf{v}^1, \frac{\omega^2}{\|\eta\|_2}\mathbf{v}^2, \ldots, \frac{\omega^s}{\|\eta\|_2}\mathbf{v}^s\right]^T \in \mathbb{R}^{sn}$ and replace $[\hat{\mathbf{B}}^{1^T}, \hat{\mathbf{B}}^{2^T}, \ldots, \hat{\mathbf{B}}^{s^T}]^T \in \mathbb{R}^{sn \times N}$ with Φ. Finally, the classification scheme based on CovSVDK features can be derived from Eq. (6) as

$$\alpha = \arg\min_{\alpha} \|\alpha\|_1 \quad \text{subject to} \quad \Phi\alpha = \mathbf{y}, \tag{10}$$

where α is the universal sparse code for representing the $\frac{\omega^i}{\|\eta\|_2}\mathbf{v}^i$ over $\hat{\mathbf{B}}^{i^T}$ for all $i = 1, \ldots, s$. Limited by space, the robustness of CovSVDK features is demonstrated in Appendix B.

3.2. Kernelizing Sparse Representation for Classification

The discrimination capability of SRC relies on the quality of the dictionary. In other words, the atoms associated to different classes must be distinguishable or separable from the perspective of ℓ_1 minimization algorithms. In some real-world applications, however, computing the sparse representation over a dictionary of original training features can yield undesirable classification results. One such example is the Iris dataset (from UCI machine learning archive). As is commonly used for analyzing the performance of various classifiers, two features for each

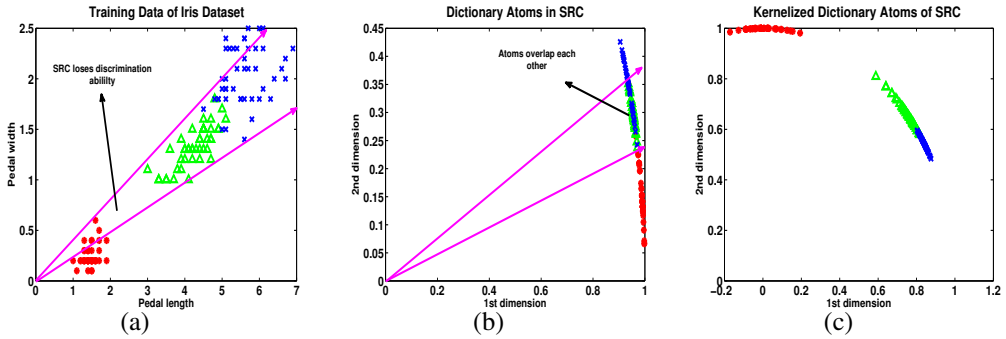

Figure 1: Training samples and dictionary atoms of SRC.

sample, regarding pedal length and pedal width, are extracted and formed into a 2D feature vector, as shown in Fig. 1(a). The three classes (points in red, green and blue) are distributed closely along the same radius direction. Obviously, the extracted 2D feature vectors are sufficiently discriminative for traditional classifiers, e.g., k-Nearest-Neighbors (kNN) and Support Vector Machines (SVMs). On the other hand, SRC normalizes training samples with unit ℓ_2-norm and employs the normalized training samples as dictionary atoms[1]. As shown in Fig. 1(b), the atoms are located on the unit circle with severe overlapping in the middle of the point scatter. The atoms within the overlapping region are inseparable and consequently cause ℓ_1 minimization algorithms the confusion in selecting the true atoms. Thus, SRC neglects the magnitude information and suffers the drawback of losing its discrimination capability in classifying data that are distributed along the same radius direction [17, 19].

We propose the kernelized sparse representation to overcome this shortcoming

[1]Normalization is typically performed to avoid trivial solution and is reasonable in face recognition, since images of a subject under different intensity levels are still considered to be same-class. In other words, the magnitudes of feature vectors are not considered as discriminative information in face recognition.

of SRC. This is desirable since by kernelizing sparse representation, the classification strategy of SRC can be applied to general pattern recognition tasks including MTS gesture recognition, time series classification, etc.

Kernel trick is a widely applied technique in machine learning that can adapt linear algorithms to nonlinear cases, by mapping training features $\phi(\cdot)$ from the original space $\mathcal{X}$ into some kernel space $\mathcal{F}$, in which the new kernel features $\psi(\cdot)$ are more separable for a certain type of classifiers and the nonlinear relationships among $\phi(\cdot) \in \mathcal{X}$ can be transformed into linear ones among $\psi(\cdot) \in \mathcal{F}$.

Let $\Psi = [\psi(\mathbf{t}_{1,1}), \ldots, \psi(\mathbf{t}_{i,1}), \ldots, \psi(\mathbf{t}_{i,n_i}), \ldots, \psi(\mathbf{t}_{k,n_k})]$ be the collection of training kernel features in $\mathcal{F}$. Given a test sample $\mathbf{x}$, we want to solve the sparse representation α of $\psi(\mathbf{x})$ over Ψ. However, this is typically infeasible, as 1) usually the mapping ψ is implicit, meaning that direct evaluation of the fitness term $\psi(\mathbf{x}) = \Psi\alpha$ is impossible [17]; 2) $\mathcal{F}$ may be of infinite dimension, causing that the computational complexity is intractable; 3) even though we know the mapping explicitly, $\Psi^T\Psi$ may not be invertible, resulting that the left inverse does not exist and thus no explicit solution to $\psi(\mathbf{x}) = \Psi\alpha$ is available. To overcome these difficulties, we introduce a relaxation to the fitness constraint term as

$$\left\| \begin{bmatrix} \psi(\mathbf{x}) \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \Psi \\ \gamma\mathbf{I} \end{bmatrix} \alpha \right\|_2 \leq \varepsilon \tag{11}$$

where $\mathbf{0} \in \mathbb{R}^N$ is a zero vector, $\mathbf{I} \in \mathbb{R}^{N \times N}$ is the identity matrix, ε is an arbitrarily small positive constant representing the error tolerance, γ is a small positive constant. Satisfying Eq. (11) is equivalent to minimizing the ridge regression problem $L(\alpha) = \|\psi(\mathbf{x}) - \Psi\alpha\|_2^2 + \gamma\alpha^T\alpha$. Setting the gradient of $L(\alpha)$ with respect to α

equal to zero, the solution space of α is obtained as

$$\Psi^T \psi(\mathbf{x}) = (\Psi^T \Psi + \gamma \mathbf{I})\alpha \tag{12}$$

where $\Psi^T \psi(\mathbf{x})$ is an $N \times 1$ vector and $\Psi^T \Psi$ is an $N \times N$ positive semi-definite matrix. Regularized by γ, $(\Psi^T \Psi + \gamma \mathbf{I})$ is invertible, yielding that α is the global minimizer to $L(\alpha)$. In other words, enabling Eq. (12) is equivalent to satisfying Eq. (11). Thus, we can employ Eq. (12) as the fitness constraint in sparse coding[2].

To improve the efficiency in ℓ_1 minimization and to ensure the solution to be sparse, a random matrix $\mathbf{P} \in \mathbb{R}^{d \times N}$ obeying Gaussian or Bernoulli distribution (we use Gaussian here) is often employed to project vector $\Psi^T \psi(\mathbf{x})$ and columns in $(\Psi^T \Psi + \gamma \mathbf{I})$ into some d-dimensional random subspace, where $d \ll N$.

Define the $\mathbf{K} = \Psi^T \Psi$ as a Gram matrix, with elements $\mathbf{K}_{i,j} = k\big(\phi(\mathbf{t}_i), \phi(\mathbf{t}_j)\big)$, where $k(\cdot, \cdot)$ is a valid kernel function. By denoting $\tilde{\mathbf{y}} = \Psi^T \psi(\mathbf{x}) = k(\cdot, \mathbf{x}) \in \mathbb{R}^N$ and substituting $\mathbf{K}$ for $\Psi^T \Psi$ in the new fitness constraint Eq. (12), the kernelized sparse representation under random projection $\mathbf{P}$ is formulated as:

$$\alpha = \arg\min_{\alpha} \|\alpha\|_1 \quad \text{subject to} \quad \mathbf{P}(\mathbf{K} + \gamma \mathbf{I})\alpha = \mathbf{P}\tilde{\mathbf{y}}. \tag{13}$$

From Eq. (13), we can see that the linear relationship between kernel features $\psi(\mathbf{x})$ and columns in Ψ has been depicted entirely in terms of the linear combination between the kernel function values in vector $\tilde{\mathbf{y}}$ and the corresponding ones in matrix $\mathbf{K}$. For the purpose of effectively classifying MTS gestures and time series

[2]Note that the proposed relaxation to fitness constraint (Eq. (11) and Eq. (12)) is a general strategy and is applicable to kernelizing other sparse coding algorithms, such as Orthogonal Matching Pursuit (OMP), but in this paper we only focus on ℓ_1 minimization algorithms.

data, we further propose two kernel functions based on the CovSVDK features.

Proposition 1 (Kernel Function). *Let* $\boldsymbol{t}$ *and* $\boldsymbol{p}$ *be two samples and let* $\phi(\boldsymbol{t})$ *and* $\phi(\boldsymbol{p})$ *be their extracted feature vectors. The proposed kernel function is defined as*

$$k\big(\phi(\boldsymbol{t}), \phi(\boldsymbol{p})\big) = \exp\Big\{k_L\big(\phi(\boldsymbol{t}), \phi(\boldsymbol{p})\big)\Big\} = \psi(\boldsymbol{t})^T\psi(\boldsymbol{p}) \tag{14}$$

where $\psi(\boldsymbol{t}) \in \mathcal{F}$ *and* $\psi(\boldsymbol{p}) \in \mathcal{F}$ *are kernel features for* $\boldsymbol{t}$ *and* $\boldsymbol{p}$*, via some implicit nonlinear mapping* ψ*. In particular, for MTS data,* $\phi(\boldsymbol{t})$ *and* $\phi(\boldsymbol{p})$ *are extracted according to Definition 1 and the kernel function* $k_L(\cdot,\cdot)$ *can be written as*

$$k_L\big(\phi(\boldsymbol{t}), \phi(\boldsymbol{p})\big) = \phi(\boldsymbol{t})^T\phi(\boldsymbol{p}) = \sum_{i=1}^{s}\left(\frac{\lambda_i\omega_i}{\|\rho\|_2\|\eta\|_2}\right)\boldsymbol{u}_i^T\boldsymbol{v}_i. \tag{15}$$

Note that kernel features $\psi(\cdot) \in \mathcal{F}$ are of infinite dimension. By working directly on the kernel function however, we can implicitly exploit the kernel space of high, or even infinite dimension, without the need of knowing mapping ψ. By using the proposed kernel function $k(\cdot,\cdot)$, the atoms embedded in a 2D random subspace for the Iris dataset are separable for ℓ_1 minimization algorithms, as shown in Fig. 1(c).

By incorporating the classification rule of SRC into Eq. (13), we obtain the newly proposed classifier, called Kernelized SRC, which shall be discussed in the following two sections.

3.3. Training

Building a discriminative dictionary is critical to the effectiveness of sparse representation based classifiers. Given a training set $\mathbf{T}$, we now describe how to construct such a dictionary via kernel trick based on specific feature extraction methods. To elaborate, we first use median filter to preprocess each sample (in the

noisy case). Then we loop through all training samples to compute the features. For MTS data, the CovSVDK feature is extracted individually from each training sample. For the case of univariate time series data, we simply employ each raw time series as a feature vector, since CovSVDK is effective only when $n \geq 2$. Next, we construct a dictionary as the regularized kernel matrix $\mathbf{K} + \gamma\mathbf{I}$. Finally, we may employ a random matrix $\mathbf{P}$ to improve the efficiency in classification. The whole training process is summarized in Alg. 1.

Algorithm 1 Kernelized SRC: Training

Require: Training set $\mathbf{T}$

Preprocess each training sample with median filter (optional)
for $i = 1$ to k **do**
  for $j = 1$ to n_i **do**
    Feature extraction for each $\mathbf{t}_{i,j} \rightarrow \phi(\mathbf{t}_{i,j})$
    (for MTS data, $\phi(\mathbf{t}_{i,j})$ is extracted according to Definition 1)
  end for
end for
Compute $\mathbf{K}$ according to Proposition 1
Construct dictionary as $\mathbf{K} + \gamma\mathbf{I}$
Secure sparsity in the solution vector by employing $\mathbf{P}$ for dimensionality reduction (optional)
return $\mathbf{P}$ and $\mathbf{P}(\mathbf{K} + \gamma\mathbf{I})$

3.4. Classification

In this section, we discuss how to classify a query sample using the proposed Kernelized SRC. Having $\mathbf{x}$ as a test sample, we first preprocess it with the same technique as in training and extract its feature as $\mathbf{y} = \phi(\mathbf{x})$. Then based on the kernel function defined in Proposition 1, we have $\tilde{\mathbf{y}} = k(\cdot, \mathbf{x}) = [k(\phi(\mathbf{t}_1), \phi(\mathbf{x})), \ldots, k(\phi(\mathbf{t}_N), \phi(\mathbf{x}))]^T \in \mathbb{R}^N$. Next, random projection can be performed to reduce dimensionality. Then, we find the sparse representation α of $\tilde{\mathbf{y}}$ over $\mathbf{P}(\mathbf{K} + \gamma\mathbf{I})$

by solving the optimization problem Eq. (13), which is called Basis Pursuit Denoising (BPD) [20]. Notice that the sparse coefficients, α, can be computed by other fast iterative algorithms, such as Orthogonal Matching Pursuit [21] or Compressive Sampling Matching Pursuit [22]. Experimental results reported in the following sections are based on the the ℓ_1 Magic implementation of BPD [23]. Finally, we identify $\mathbf{x}$ as class i based on the decision rule as:

$$i = \arg \min_{i \in \{1,\ldots,k\}} \|\mathbf{P}\tilde{\mathbf{y}} - \mathbf{P}(\mathbf{K} + \gamma\mathbf{I})\delta_i(\alpha)\|_2, \tag{16}$$

where $\delta_i(\alpha) = [0, \ldots, \alpha_{i,1}, \ldots, \alpha_{i,n_i}, \ldots, 0]$. To cope with unbalanced classes, an alternative decision rule $i = \arg\min_{i \in \{1,\ldots,k\}} \frac{\|\mathbf{P}\tilde{\mathbf{y}} - \mathbf{P}(\mathbf{K}+\gamma\mathbf{I})\delta_i(\alpha)\|_2}{\|\delta_i(\alpha)\|_1}$ can be employed. The classification procedure is summarized in Alg. 2.

Algorithm 2 Kernelized SRC: Classification

Require: Test sample $\mathbf{x}$, random matrix $\mathbf{P}$ and dictionary $\mathbf{P}(\mathbf{K} + \gamma\mathbf{I})$

1: Preprocess test sample with median filter (optional)
2: Feature extraction for $\mathbf{x} \rightarrow \mathbf{y} = \phi(\mathbf{x})$ according to Definition 1
3: Based on the kernel function defined in Proposition 1, compute $\tilde{\mathbf{y}} = k(\cdot, \mathbf{x}) = [k(\phi(\mathbf{t}_1), \phi(\mathbf{x})), \ldots, k(\phi(\mathbf{t}_N), \phi(\mathbf{x}))]^T$
4: Random subspace embedding via $\mathbf{P}$ (optional)
5: Find the sparse coefficient vector α by solving Eq.(13)
6: $i = \arg \min_{i \in \{1,\ldots,k\}} \|\mathbf{P}\tilde{\mathbf{y}} - \mathbf{P}(\mathbf{K} + \gamma\mathbf{I})\delta_i(\alpha)\|_2$
7: **return** i

4. Experiments on Classifying Real-World MTS Data

In this section, we conduct experiments to demonstrate the promising performance of the proposed framework, *i.e.*, CovSVDK + Kernelized SRC, over three online public-access databases, *i.e.*, the Georgian-Tech Human Gait (Georgia-

Tech HG) database[1], Australian Sign Language (Auslan) database[2] and High-quality Australian Sign Language (HAuslan) database[2]. The Georgia-Tech HG database was obtained via 12 video cameras; the Auslan was generated by Powergloves; and the HAuslan was generated by two 5DT gloves and two position trackers. To verify the effectiveness of the proposed CovSVDK feature, we use the linear kernel $k_L(\cdot,\cdot)$ for all experiments in this section. Feature vector $\phi(\cdot)$ for each MTS is extracted according to Definition 1. For each particular database, the parameter s is manually selected and is consistent for all MTS data within the database. As in [2], atoms in $\mathbf{P}(\mathbf{K}+\gamma\mathbf{I})$ are normalized to unit ℓ_2-norm prior to ℓ_1 minimization. γ is set to 0.001.

We evaluate and compare the proposed CovSVDK, with Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA). For PCA and LDA, all MTS data are interpolated or downsampled to the average length, in each database. We compare the proposed classifier Kernelized SRC with two popular classifiers, *i.e.,* K-Nearest-Neighbor (KNN) with $k = 3$, Support Vector Machines (SVM) and with the coding strategy by computing the least square solution to Eq. (12), termed LS. For Kernelized SRC and LS, the decision rule is Eq. (16). For KNN and SVM, columns in Φ are employed as training data and $\phi(\mathbf{x})$ is used as the test sample. The SVM toolbox can be found at [24]. As shown in the following, our method consistently achieves high performance over these databases.

[1]Published by the Computational Perception Laboratory at Gatech at http://www.cc.gatech.edu/cpl/projects/hid/

[2]Published by UCI KDD at http://kdd.ics.uci.edu/summary.data.date.html

4.1. Georgia-Tech HG database

The Georgia-Tech HG database, used for human identification from a distance, is a collection of human gaits from 15 subjects. Samples of subjects were captured by cameras at 4 different controlled speeds [9]. Every subject was required to walk 9 times at every controlled speed and finally, 36 samples were obtained for every subject. A sample is a time series of gaits with varying length. By means of 22 markers on the subject, a gait is defined by 66 attributes (variables), *i.e.*, the 3-D coordinates of those markers [25, 26]. The evaluation uses all the 540 samples in the database. Among the 36 samples per subject, 30 samples are randomly collected into the training set while the remaining 6 samples are used for testing.

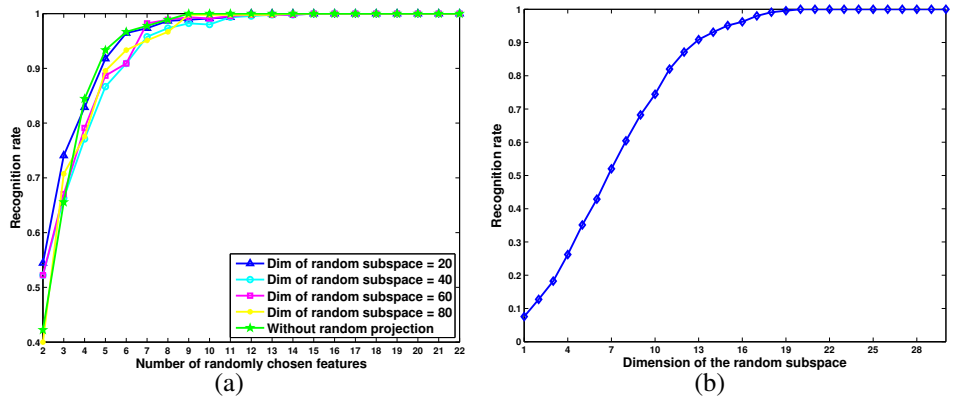

Figure 2: Recognition rate for the Georgia-Tech HG database. (a) 15-class problem recognition rate versus selected features (markers) under various random projections. The horizontal axis represents the number of randomly chosen features, ranging from 2 to 22. The curves in different colors represents recognition rate over 5 different random subspaces. (b) 15-class problem recognition rate versus dimensions of random subspace; 22 features (markers) are employed.

By transforming the kernel matrix into a low dimensional random subspace, we can reduce the computation cost of ℓ_1 minimization. In order to evaluate the effectiveness of random projection, we randomly select parts of the overall 22 markers and set the parameter $s = 5$ uniformly, such that 5 singular value/vector pairs are extracted by CovSVDK for each MTS. Figure 2(a) indicates that the

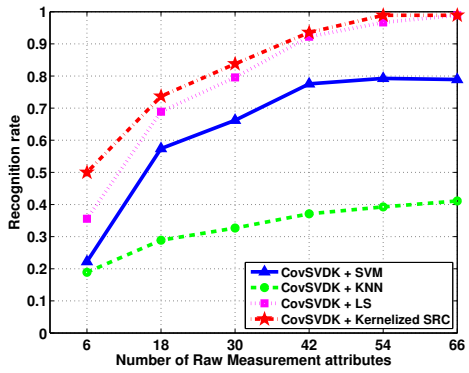

Figure 3: Recognition rate for various methods over the Georgia-Tech HG database.

proposed approach can achieve 100% recognition rate when a random subspace is of only 20 dimensions and only 11 markers are utilized. Hence, in the following experiments over this database, kernel matrices are projected onto a random subspace with dimension 20 to improve computation efficiency.

Remark: It is worthy to point out that, for ℓ_1 minimizers, the dimensionality reduction induced by random projection is not a requisite. The purpose of embedding the dictionary atoms into some low-dimensional subspace is two-fold: 1) speed-up ℓ_1 minimization; 2) enforce the dictionary to be overcomplete such that the solution tends to be sparse. The first concern is desired from a practical efficiency perspective while the second concern is preferred by the decision rule (Eq.(16)) so as to secure satisfactory recognition rate. We can see from Figure 2(b) that the recognition rate increases as the dimension of the random subspace becomes higher. For completeness, we also evaluate the proposed approach over the Georgia-Tech HG database without performing dimensionality reduction. Figure 2(a) illustrates that the accuracy obtained without dimensionality reduction is similar to those with dimensionality reduction.

To evaluate the proposed framework in a more challenging scenario, we down-

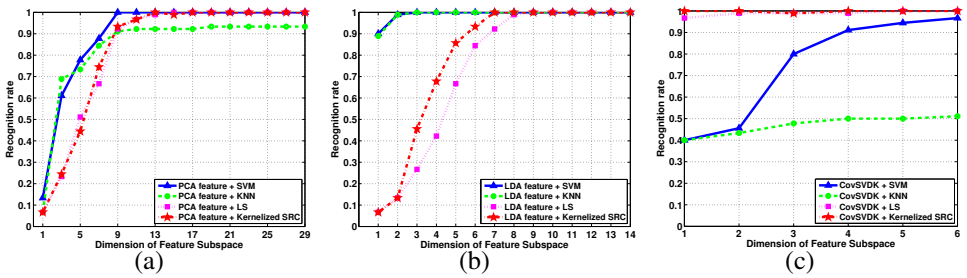

Figure 4: Recognition rate on the Georgia-Tech HG database. (a) PCA feature (b) LDA feature (c) CovSVDK feature (proposed method). All three feature extraction methods are fed to four classifiers, i.e., SVM, KNN, LS, the proposed Kernelized SRC.

Database	Proposed k_L	Exponential	Poly.(d = 3)	Gaussian
Gait	100%	92.2%	85.6%	80.4%

Table 1: Comparison among different kernel functions over the Georgia-Tech HG database.

sample the raw gesture data into $1/5$ of its original length and utilize only part of the overall 66 attributes. As shown in Figure 3, our method robustly achieves 98.9% recognition, leading SVM by approximately 10% in accuracy.

As shown in Figure 4, at 9, 4, and 1 dimension(s) of the feature subspace respectively, PCA, LDA and CovSVDK achieve 100% recognition rate. Therefore, compared with PCA and LDA, the proposed CovSVDK is more effective in preserving discriminative information for classification. Finally, Table 1 shows that in classifying MTS data, the proposed linear kernel function $k_L(\cdot,\cdot)$ significantly outperforms three other popular kernel functions, *i.e.*, exponential, polynomial and Gaussian kernel functions.

4.2. Australian Sign Language (Auslan) database

Contributed by 5 individual signers, the Auslan database contains 95 one-hand signs. 70 samples were collected for each sign and a sample is comprised of varying-length time series for a single hand gesture. There are 15 attributes or

features for each gesture, *i.e.*, the x, y and z coordinates of the palm, the angles (roll, pitch and yaw) of the palm, the bend values of the 5 fingers and 4 additional setting values. Over this database, we conduct comparative study by evaluating the proposed approach (CovSVDK + Kernelized SRC) against several state-of-the-art algorithms, *i.e.,* discriminative mixture learning (MixCML [9]), Dynamic Time Warping (DTW) [11], Fourier Descriptors [27] and SRC [2]. Recognition rates are cited from literature for the first three methods. Results for SRC are reported based on our own implementation.

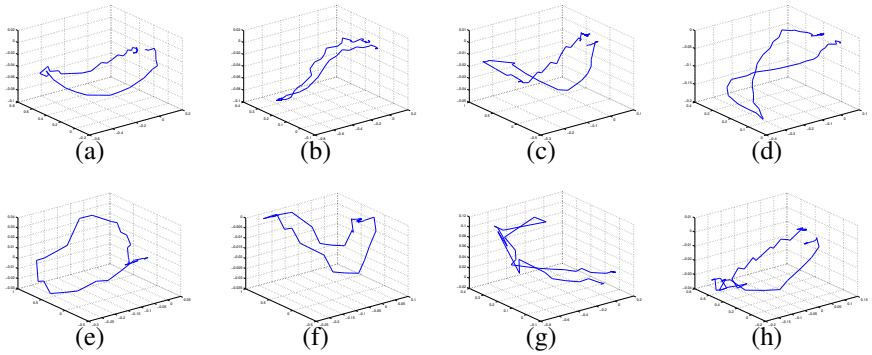

Figure 5: 3D trajectories for 8 signs. (a) Eat, (b) Exit, (c) Forget, (d) Give (e) Hello, (f) Know, (g) Love (h) No.

In the first experiment over the Auslan database, we consider a binary classification task. With the same experiment setup as [9], we form a subset by using 10 signs and choose, from the 15 attributes, 8 attributes, namely the x, y and z coordinates of the palm, the roll angle of the palm, the bend values of the fingers of thumb, fore, index and ring.

For each of the 10 signs, *i.e.*, "eat", "exit", "forget", "give", "hello", "know", "love", "no", "sorry" and "yes", we select approximately 4 samples from each signer. Conducting 10-fold cross-validation yields a training set of 36 samples (18 per sign) and a test set of 4 samples. The proposed framework is compared

with MixCML [9] and DTW [11], and the results are listed in Table 2. For completeness, the proposed method is further examined by performing binary classification over various selection of attributes. The results are summarized in Table 3. Consistent with the argument made by Kim and Pavlovic [9], our observation also reveals that the $7^{th} - 10^{th}$ attributes are less discriminative than others as they only provide the finger flexion information.

Method	Training Set	Test Set	Recognition Rate
Proposed	36	4	96.3%
MixCML [9]	39	1	$95.5\%*$
DTW [11]	39	1	$88\%*$

Table 2: Binary Classification comparison among various methods over the Auslan database. Recognition rates with $*$ are cited from [9].

Method	Selected Attributes	Recognition Rate
Proposed	$1^{th} - 4^{th}$, $7^{th} - 10^{th}$	96.3%
	$1^{th} - 6^{th}$	94.5%
	$1^{th} - 4^{th}$	96.3%
	$7^{th} - 10^{th}$	70.0%
	$1^{th} - 3^{th}$	96.3%

Table 3: Binary classification result over the Auslan database for various selection of attributes.

In literature, we notice that this database has been widely applied to evaluate spatial trajectory recognition algorithms. In the second experiment, for fair comparison, we only keep 3 attributes, *i.e.*, x, y and z coordinates. Figure 5 gives some examples for 8 signs. Using the same CovSVDK features, we first compare two classifiers *i.e.,* Kernelized SRC and SRC based on 10-fold cross-validation. Then keeping the experiment setup consistent as in [27], the proposed approach (CovSVDK + Kernelized SRC) is compared with DTW [11] and Fourier Descriptor [27] based on 2-fold cross-validation. Classification results for aforementioned

methods are summarized in Table 4, which indicates that the proposed algorithm is competitive among these advanced trajectory recognition algorithms.

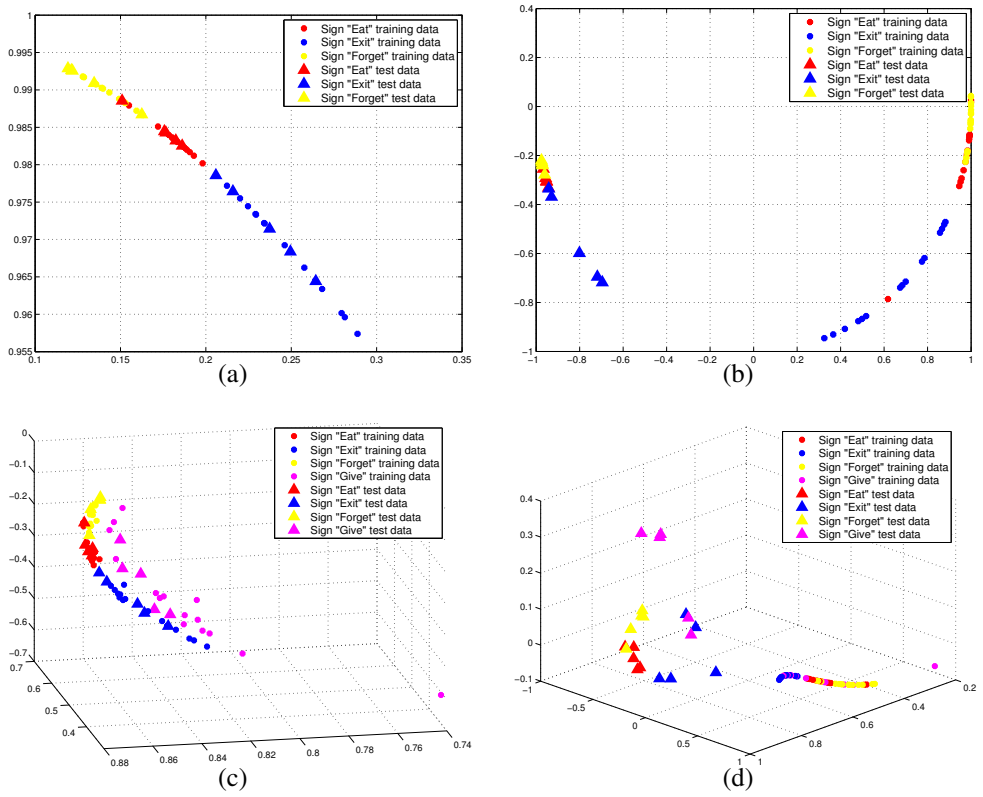

Figure 6: Illustrations of manifolds in multi-class classification tasks. Top row: the 3-label task; bottom row: the 4-label task. (a) 2D manifold with the kernel trick, (b) 2D manifold without the kernel trick, (c) 3D manifold with the kernel trick, (d) 3D manifold without the kernel trick.

The effectiveness of Kernelized SRC is illustrated in Figure 6, in which, for better visualization, 15 samples per sign are utilized for training while the remaining 5 samples are for testing. The 2/3D manifolds are obtained by projecting the dictionaries (with and without the kernel trick) into random subspace. Clearly, with kernel trick, samples from different classes are more separable than those without the kernel trick, which reveals that the proposed classifier is more robust

Method	Train set : Test set	Classes			
		2	3	4	8
Proposed 1	$0.9 : 0.1$	96.3%	93.3%	90.6%	80.0%
SRC [2]	$0.9 : 0.1$	78.5%	73.3%	70.9%	63.0%
Proposed 2	$0.5 : 0.5$	96.0%	92.7%	88.0%	75.4%
DTW [11]	$0.5 : 0.5$	$89.8\%*$	N/A	$83.8\%*$	$75.9\%*$
Fourier Descriptor [27]	$0.5 : 0.5$	$82.1\%*$	N/A	$63.7\%*$	$52.3\%*$

Table 4: Multi-class Classification comparison among various methods over the Auslan database. Recognition rates with $*$ are cited from [27]. Proposed 1 is based on 10-fold cross-validation; For proposed 2, the data pool is divided into 2 folds, *i.e.*, one fold for training and the other fold for test, according to [27].

than SRC [2] when dealing with cluttered data.

4.3. High-quality Australian Sign Language (HAuslan) database

The HAuslan database consists of 95 two-hand signs. Compared with the Auslan database, the number of samples per sign is reduced to 27 and the number of attributes is increased to 22, (11 attributes for each hand). The 11 attributes for one hand are the same as those in Auslan database excluding the 4 setting values.

First, to illustrate the capability of our method in classifying large-scale databases, all 95 sign classes are used. Since the HAuslan database contains much more classes but fewer samples per class than previous two databases, 24 randomly selected samples are assigned to training set for each sign, while the remaining 3 samples are collected into the test set. Note that the kernel matrix contributed by all training samples is of size 2280×2280, to which performing ℓ_1 minimization is computationally expensive. For efficient classification, we employ random projection to reduce the row dimension of the kernel matrix to 40, which is just 1.8% of its original size. In addition, considering that the subtle differences among some signs, we set $c = 99.9\%$ so as to involve sufficient gesture details to enable effective classification. To improve robustness and remove outlier atoms from the

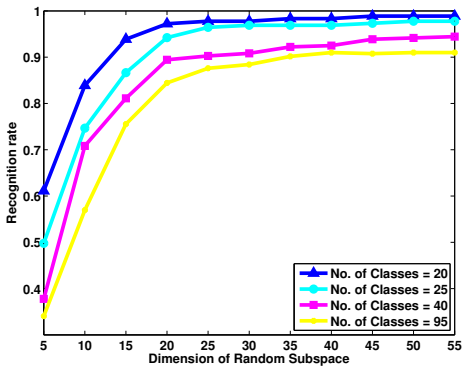

Figure 7: Recognition rate for the HAuslan Database.

dictionary, we apply a refinement process to the dictionary by only preserving the atoms with large reconstruction coefficients, based on the solution to Eq. (13). Then, the newly formed sub-dictionary is fed to the classifier. The recognition rates of the proposed framework (CovSVDK + Kernelized SRC) are presented in Table 5 and in Figure 7.

Classes:samples	20:540	25:675	40:1080	95:2565
Recognition rate	98.2%	97.6%	94.3%	91.2%

Table 5: Recognition rate on the HAuslan database. The dimension of random subspace is fixed at 40 for all the classification tasks.

Next, we compare CovSVDK + Kernelized SRC with various combinations of feature extraction strategies and classifiers. For CovSVDK, we set the parameter $s_{max} = 6$ and for PCA, we set the energy preservation ratio $c_{max} = 99.9\%$, which results in a maximal 30 features. The maximal number of linear features for LDA is 21. Figure 8 shows that although Kernelized SRC using PCA and LDA features yields inferior performance to SVM[3], when working jointly with CovSVDK,

[3]This is due to the fact that Kernelized SRC uses the simplest linear kernel while SVM employs the more advanced RBF kernel.

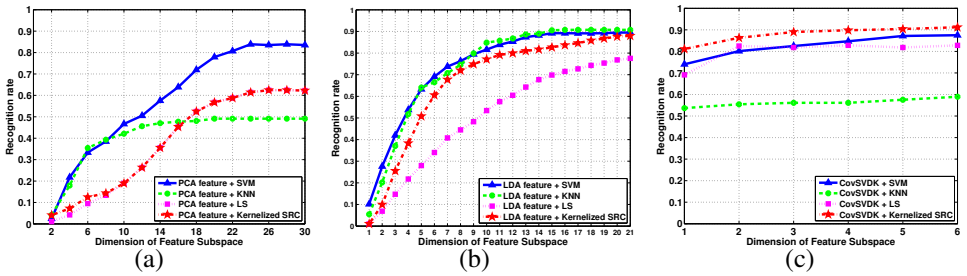

Figure 8: Recognition rate over the HAuslan database. (a) PCA feature (b) LDA feature (c) CovSVDK feature (proposed method). All three feature extraction methods are fed to four classifiers, i.e., SVM, KNN, LS, the proposed Kernelized SRC.

Methods	Proposed	PCA+SVM	LDA+SVM	LDA+KNN
Features	6	28	18	18
Accuracy	91.2%	83.4%	90.0%	90.4%

Table 6: Summary of recognition performance on the HAuslan database.

Database	Proposed k_L	Exponential	Poly. (d = 3)	Gaussian
HAuslan	91.2%	76%	75.8%	78.9%

Table 7: Comparison among different kernel functions over the HAuslan database.

Kernelized SRC outperforms other combinations of features and classifiers. This result confirms the effectiveness of the proposed framework. The highest recognition rates and the corresponding dimensions of feature space for various methods are summarized in Table 6. As shown in Table 7, in classifying MTS data, the proposed kernel function $k_L(\cdot,\cdot)$ again significantly outperforms three other widely used kernel functions, *i.e.*, exponential, polynomial and Gaussian kernel functions.

Finally, a comparison among state-of-the-art methods in the 25-label classification problem is given in Table 8, which further validates the superiority of the proposed method.

Method	Proposed	Li [7]	2dSVD [28]	SegSVD [29]
Accuracy	97.6%	89.0%*	95.0%*	93.9%*

Table 8: Comparison of recognition rate among various methods over the HAuslan database. Recognition rates with $*$ are cited from references.

4.4. Evaluating the Robustness

In this section, we evaluate the robustness of the proposed framework by employing the Sparsity Concentration Index (SCI) [2] to detect outliers. The SCI is defined as [2]

$$SCI(\alpha) = \frac{k \cdot \max_i \|\delta_i(\alpha)\|_1 / \|\alpha\|_1 - 1}{k - 1}, \tag{17}$$

where α is the solution to Eq. (13) and $\delta_i(\alpha)$ is the characteristic function defined in Eq. (16). If a test sample can be entirely expressed by the training samples from only a single class, then $SCI(\alpha) = 1$; while, in the other extreme, if the coefficients in α spread evenly over the classes, then $SCI(\alpha) = 0$. The intuition lies in the fact that, for a test sample belonging to a certain class in the training set, the large sparse coefficients should be mostly concentrated on the same-class training samples and therefore yield an SCI that approaches 1. On the other hand, if the test sample is an irrelevant outlier, then its sparse coefficients should spread almost evenly across the whole training set and yield an SCI close to 0. Thus, the outlier detection criterion [2] is established, by setting a threshold $\tau \in (0, 1)$, where a test sample is rejected as outlier if $SCI(\alpha) < \tau$.

We verify the robustness of the proposed method over the Georgia-Tech HG and the HAuslan databases. As recommended in [2], we incorporate approximately half of all the classes into the training set but keep the test set containing samples from all the classes. Thus almost half of the test set are considered as irrelevant outliers with respect to the dictionary. For the two databases, the number

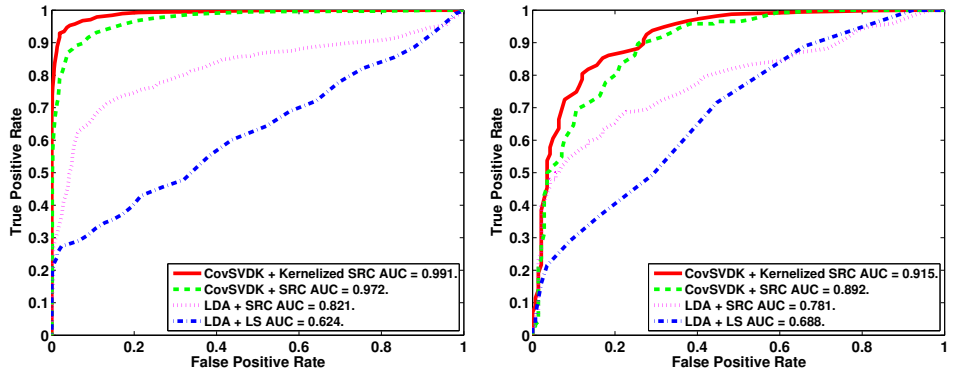

Figure 9: ROC curves for outlier detection over the Georgia-Tech HG and the HAuslan databases. (a) the Georgia-Tech HG database, (b) the HAuslan database. CovSVD means feature extraction following Definition. 1 and Definition. 2, before applying the kernel trick.

of classes employed in the training set are 8 and 48 respectively. We test the performance of the proposed algorithm (CovSVDK + Kernelized SRC) by ranging τ from 0 to 1 with 0.01 step size. The resultingt Receiver Operator Characteristic (ROC) curves, (Figure 9), indicate that: 1) the proposed CovSVDK outperforms classical LDA in outlier detection; 2) Kernelized SRC demonstrates improved robustness compared to SRC; and 3) the Area Under Curve (AUC) of the proposed framework exceeds the AUC of other listed approaches.

5. Experiments on Classifying Real-World Univariate Time Series Data

In this section, we evaluate the proposed classifier Kernelized SRC with nonlinear kernel function $k(\cdot,\cdot)$ over 20 datasets (data1) from UCR Time-Series Repository [30]. Raw time series are directly treated as feature vectors $\phi(\cdot)$ without using CovSVDK, which is effective only when $n \geq 2$[4]. The regularization param-

[4]To avoid the similarity values out of range, a normalizing $\phi(\cdot)$ to unit ℓ_2-norm or dividing the matrix entries by N is needed. We choose the former strategy in this work.

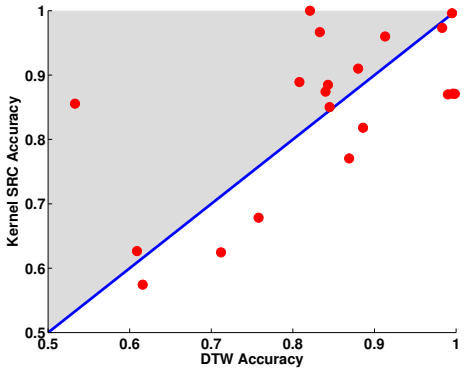

Figure 10: Accuracy scatter plot between Kernelized SRC and 1NN-Best Warping Window DTW [30]. Each dot represents a dataset. Dots above the diagonal mean that Kernelized SRC is better than 1NN-Best Warping Window DTW and vice versa. The farther away a dot is from the diagonal, the greater the accuracy improvement achieved [13].

eter γ is set to 0.001. All columns in $\mathbf{P}(\mathbf{K} + \gamma\mathbf{I})$ are normalized to unit ℓ_2-norm prior to sparse coding. The dictionary employed is the kernel matrix with compression rates $\{\frac{d}{N} = 0.10, 0.25, 0.50, \text{none}\}$ induced by random projection, where none means no dimensionality reduction. The best result from the four cases is reported.

We compare Kernelized SRC with state-of-the-art time series classifiers, *i.e.,* 1NN-Best Warping Window DTW [12], Time Series based on a Bag-of-Features representation (TSBF) [31], as well as 7 classic classifiers[5]. The error rates of all methods are listed in Table 9, from which we can see that Kernelized SRC leads other algorithms by yielding the lowest error rate in 7 out of the 20 datasets. In particular, we visualize the accuracy scatter plot between Kernelized SRC and 1NN-Best Warping Window DTW [12], which is considered one of the best time series classifiers. As shown in Figure 10, the proposed classifier slightly outperforms 1NN-Best Warping Window DTW in 11 out of 20 datasets.

[5]The information regarding classic machine learning algorithms is summarized in http://www.cs.ucr.edu/~eamonn/time_series_data/WekaOnTimeSeries.xls

In addition, to fully justify the effectiveness of the proposed kernelization strategy, we test SRC over the 20 datasets and compare it with Kernelized SRC by visualizing the accuracy scatter plot. Figure 11(a) shows that using kernel trick significantly improves the classification performance, as Kernelized SRC outperforms SRC in 19 out of 20 datasets. Moreover, a classifier is useful only if we can predict ahead of time on which datasets it will generate higher accuracy. We therefore perform further experiments to verify the reliability of Kernelized SRC by evaluating the expected accuracy gain versus the actual accuracy gain [32]. To acquire the expected accuracy gain, we conduct leave-one-out cross-validation within the training set for both algorithms. The gain is calculated as [32] $g = \frac{\text{Accuracy Kernelized SRC}}{\text{Accuracy SRC}}$. As depicted in Figure 11(b), 19 out of the 20 dots are in region TP with the remaining 1 in region TN, which indicates that the performance of Kernelized SRC is completely predictable over the 20 datasets. From the same figure, we also observe that a remarkable 20% or even higher performance increase compared to SRC is achieved via kernelization over a majority of the datasets. The impressive results validate that the proposed Kernelized SRC is very effective for time series classification.

6. Conclusion and Future Work

In this paper, we propose a novel sparse representation based framework for classifying complicated human gestures captured as multi-variate time series (MTS). First, we propose a feature extraction strategy, called CovSVDK, which is invariant to inconsistent lengths and temporal disorder across MTS data, robust to variability within human gestures, and efficient to compute. In addition, we propose a new approach to kernelize sparse representation by introducing a relaxation to

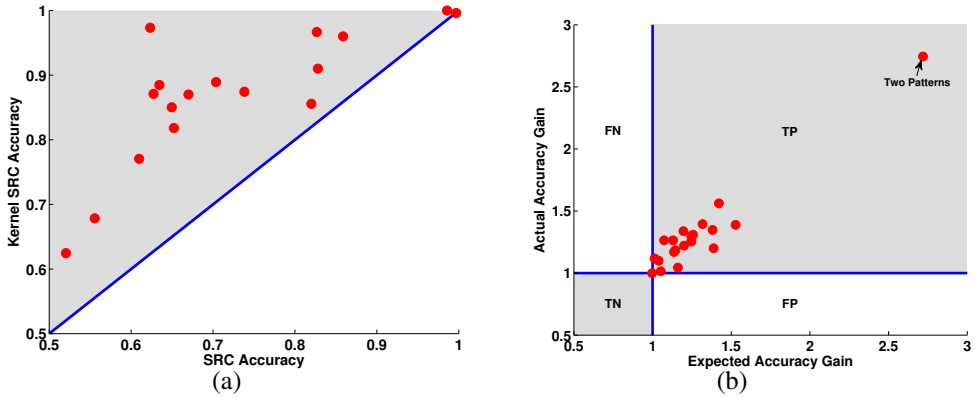

Figure 11: Comparison between Kernelized SRC and SRC. (a) accuracy scatter plot; (b) expected accuracy gain versus actual accuracy gain. Note that regions marked as TP/TN represent we correctly predict Kernelized SRC is better/worse than SRC; region FN means that we predict Kernelized SRC is worse than SRC but the fact is the opposite; region FP means that we predict Kernelized SRC is better than SRC but the fact is the opposite. Practically, only FP is the truly bad case [32].

the fitness constraint. This technique is generic and can be applied to kernelizing other sparse coding algorithms. Using this technique, we derive a classifier called Kernelized SRC, which is very effective in classifying MTS data and univariate time series as shown in the experiments.

In our future work, the proposed approach will be combined with a multi-layer structure that can model complicated temporal variations to further improve

	Knn	NB	$C4_5$	MLP	RandForest	LMT	SVM	DTW* [12]	TSBF* [31]	Kernelized SRC
50words	35.60%	43.74%	58.24%	33.63%	44.84%	43.08%	35.38%	24.20%	**19.10%**	32.16%
Adiac	40.66%	43.22%	46.80%	**25.06%**	42.20%	27.88%	56.01%	39.10%	28.60%	37.34%
Beef	40.00%	50.00%	43.33%	26.67%	50.00%	20.00%	33.33%	46.70%	35.00%	**14.44%**
CBF	15.00%	10.33%	32.67%	14.67%	16.44%	23.00%	12.33%	**0.40%**	0.50%	12.89%
Coffee	25.00%	32.14%	42.86%	3.57%	25.00%	**0.00%**	3.57%	17.90%	0.40%	**0.00%**
ECG200	11.00%	23.00%	28.00%	16.00%	19.00%	18.00%	19.00%	12.00%	13.80%	**9.00%**
FaceAll	31.36%	30.83%	44.97%	17.57%	39.05%	24.26%	28.17%	19.20%	21.70%	**11.08%**
FaceFour	12.50%	15.91%	28.41%	12.50%	21.59%	22.73%	11.36%	11.40%	**3.80%**	18.18%
Fish	21.71%	33.14%	40.00%	16.00%	20.57%	18.29%	14.86%	16.00%	**7.10%**	12.57%
Gun Point	8.00%	21.33%	22.67%	6.67%	10.67%	20.67%	20.00%	8.70%	**1.10%**	4.00%
Lighting2	19.67%	32.79%	37.70%	26.23%	21.31%	36.07%	27.87%	**13.10%**	24.90%	22.95%
Lighting7	36.99%	35.62%	45.21%	35.62%	43.84%	35.62%	**28.77%**	28.80%	30.70%	37.54%
OliveOil	23.33%	23.33%	26.67%	13.33%	13.33%	16.67%	13.33%	16.70%	11.30%	**3.33%**
OSULeaf	45.45%	62.81%	63.22%	55.37%	58.26%	50.83%	56.20%	38.40%	**23.30%**	42.56%
SwedishLeaf	20.32%	14.56%	34.40%	13.44%	22.24%	17.44%	15.84%	15.70%	**8.90%**	11.52%
Synthetic Control	12.00%	4.00%	19.00%	8.67%	14.00%	8.00%	7.67%	**1.70%**	1.90%	2.67%
Trace	18.00%	20.00%	26.00%	23.00%	19.00%	24.00%	27.00%	**1.00%**	2.00%	13.00%
Two Patterns	9.40%	54.33%	34.88%	10.35%	27.50%	16.78%	17.80%	**0.15%**	0.10%	12.92%
Wafer	0.60%	29.17%	1.80%	3.72%	0.68%	1.91%	4.04%	0.50%	0.40%	**0.38%**
Yoga	16.70%	45.77%	30.10%	25.50%	22.13%	28.13%	36.93%	15.50%	16.00%	**14.81%**

Table 9: Classification results on UCR Time-Series Repository. Note that DTW* [12] means 1NN-Best Warping Window DTW and TSBF* [31] represents Time Series based on a Bag-of-Features representation with the optimal parameter setting $z = 0.25$. Results for compared methods are cited from references.

the recognition performance. The performance of incorporating nonlinear kernel function into Kernelized SRC for classifying gesture MTS data will also be extensively investigated.

7. Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0812458.

References

[1] M. Aharon, M. Elad, A. Bruckstein, K-svd: An algorithm for designing overcomplete dictionarys for sparse representation, IEEE Transactions on Signal Processing 54 (11) (2006) 4311 – 4322. 2

[2] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210 – 227. 2, 5, 6, 8, 18, 22, 25, 28

[3] Q. Zhang, B. Li, Discriminative k-svd for dictionary learning in face recognition, IEEE Conference on Computer Vision and Pattern Recognition (2010) 2691 – 2698. 2

[4] Y. Li, C. Fermuller, Y. Aloimonos, H. Ji, Learning shift-invariant sparse representation of actions, IEEE Conference on Computer Vision and Pattern Recognition (2010) 2630 – 2637. 2

[5] K. Yang, C. Shahabi, A pca-based similarity measure for multivariate time series, MMDB' 04: Proceedings of the 2nd ACM international workshop on Multimedia databases (2004) 65 – 74. 2, 3, 5

[6] C. Li, S. Q. Zheng, B. Prabhakaran, Segmentation and recognition of motion streams by similarity search, ACM Trans. on Multimedia Computing, Communications and Applications 3 (3). 2, 5

[7] C. Li, P. Zhai, S. Zheng, B. Prabhakaran, Segmentation and recognition of multi-attribute motion sequences, Proceedings of the ACM Multimedia Conference 2004 (2004) 836 – 843. 2, 3, 5, 28

[8] K. Yang, C. Shahabi, A pca-based kernel for kernel pca on multivariate time series, Proceedings of ICDM 2005 Workshop on Temporal Data Mining: Algorithms, Theory and Applications (2005) 149 – 156. 2, 3, 5

[9] M. Kim, V. Pavlovic, Discriminative learning of mixture of bayesian network classifiers for sequence classification, IEEE Conference on Computer Vision and Pattern Recognition (2006) 268 – 275. 2, 4, 19, 22, 23

[10] Y. Yuan, K. E. Barner, Hybrid feature selection for gesture recognition using support vector machines, IEEE Conference on ICASSP (2008) 1941 – 1944. 2

[11] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, Acoustics, Speech and Signal Processing, IEEE Transactions on 26 (1) (1978) 43 – 49. 3, 4, 22, 23, 25

[12] C. A. Ratanamahatana, E. J. Keogh, Making time-series classification more accurate using learned constraints, in: SDM'04, 2004. 3, 4, 30, 32

[13] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, E. Keogh, Indexing multi-dimensional time-series with support for multiple distance measures, ACM SIGMOD (2003) 216 – 225. 3, 4, 30

[14] A. L. Vatavu, R.-D., J. Wobbrock, Gestures as point clouds: A $ p recognizer for user interface prototypes, in: ICMI, 2012. 3

[15] F. Bashir, A. Khokhar, D. Schonfeld, Object trajectory-based activity classification and recognition using hidden markov models, Image Processing, IEEE Transactions on 16 (7) (2007) 1912 –1919. 4

[16] W. J. Krzanowski, Between-groups comparison of principal components, JASA 74 (367) (1979) 703 – 707. 4

[17] L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, F.-Z. Li, Kernel sparse representation-based classifier, Signal Processing, IEEE Transactions on. 5, 12, 13

[18] S. Gao, I. Tsang, L.-T. Chia, Kernel sparse representation for image classification and face recognition, in: Computer Vision ECCV 2010. 5

[19] Y. Zhou, J. Gao, K. E. Barner, An enhanced sparse representation strategy for signal classification, in: Proceedings, SPIE Defense, Security, and Sensing, 2012. 12

[20] E. J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Info. Theory 52 (2) (2006) 489 – 509. 17

[21] J. Tropp, A. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, Information Theory, IEEE Trans. 53 (12) (2007) 4655 –4666. 17

[22] D. Needell, J. A. Tropp, Cosamp: iterative signal recovery from incomplete and inaccurate samples, Commun. ACM 53 (12) (2010) 93–100. 17

[23] E. Candès, J. Romberg, l1-magic: Recovery of sparse signals via convex programming. 17

[24] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, 2001. available at http://www.csie.ntu.edu.tw/cjlin/libsvm. 18

[25] R. Tanawongsuwan, A. Bobick, Characteristics of time-distance gait parameters across speeds, GVU Technical Report. 19

[26] R. Tanawongsuwan, A. Bobick, Performance analysis of time-distance gait parameters under different speeds, 4th International Conference on Audio and Video Based Biometric Person Authentication. 19

[27] S. Wu, Y. Li, On signature invariants for effective motion trajectory recognition, The International Journal of Robotics Research. 27 (8) (2008) 895 – 917. 22, 23, 25

[28] X. Weng, J. Shen, Classification of multivariate time series using two-dimensional singular value decomposition, Knowledge-Based Systems 21 (7) (2008) 535 – 539. 28

[29] J. Liu, M. Kavakli, Hand gesture recognition based on segmented singular value decomposition, in: Knowledge-Based and Intelligent Information and Engineering Systems, Vol. 6277, 2010, pp. 214–223. 28

[30] E. Keogh, Q. Zhu, B. Hu, H. Y., X. Xi, L. Wei, R. C. A. (2011)., The ucr time series classification/clustering homepage. available at http://www.cs.ucr.edu/eamonn/time_series_data/. 29, 30

[31] M. G. Baydogan, G. Runger, E. Tuv, A bag-of-features framework to classify time series, IEEE Transactions on Pattern Analysis and Machine Intelligence.Submitted for publication. 30, 32

[32] G. E. A. P. A. Batista, X. Wang, E. J. Keogh, A complexity-invariant distance measure for time series., in: SDM, 2011. 31, 32

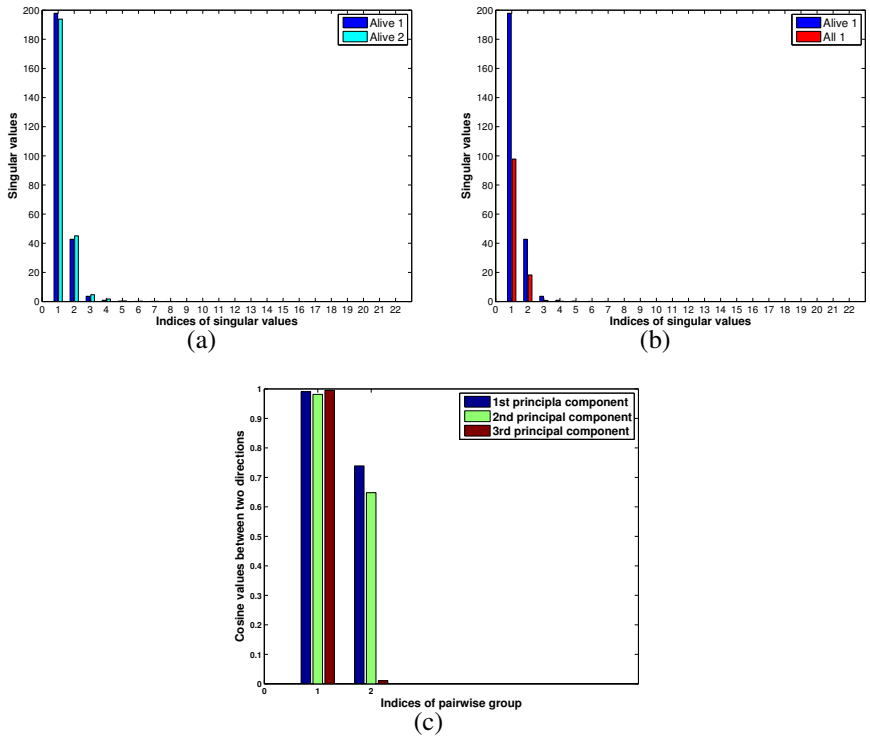

Figure A.12: Discriminative properties of SVD for MTS data from the HAuslan Database. (a) Pairwise comparison of singular values between two MTS data, both representing sign "Alive"; (b) Pairwise comparison of singular values between two MTS data, respectively representing the signs "Alive" and "All"; (c) Comparison of the directions of the singular vector pairs. Group 1 is the comparison between two MTS data from the same class "Alive" and group 2 is the comparison between two MTS data from classes "Alive" and "All" respectively.

Appendix A. SVD Properties of MTS Data

As shown in Fig. 12(a), the singular values between two same-class MTS data, both standing for the sign "Alive" from HAuslan database, resemble to each other correspondingly and shrink to zero quickly as the index growing. For the two MTS data pertinent to different classes (sign "Alive" and sign "All" from HAuslan database), the singular values differ significantly from each other, as shown in Fig. 12(b). To measure the resemblance in direction between two singular vectors, we can simply compute the cosine value of the acute angle between them defined

as

$$\cos\theta = | < \mathbf{u}_i, \mathbf{v}_i > |, \tag{A.1}$$

where $\theta \in [0, \pi/2]$ and $<,>$ is the inner product operator. As illustrated Fig. 12(c), similarity value between singular vectors (principle component) of two same-class MTS data is close to 1, meaning that their directions are similar. On the other hand, if the two MTS data are associated to different classes, the similarity value between singular vectors is significantly smaller than 1. Therefore, the singular values and singular vectors obtained via SVD provide discriminative information for classifying MTS data.

Appendix B. Effectiveness of CovSVDK Features

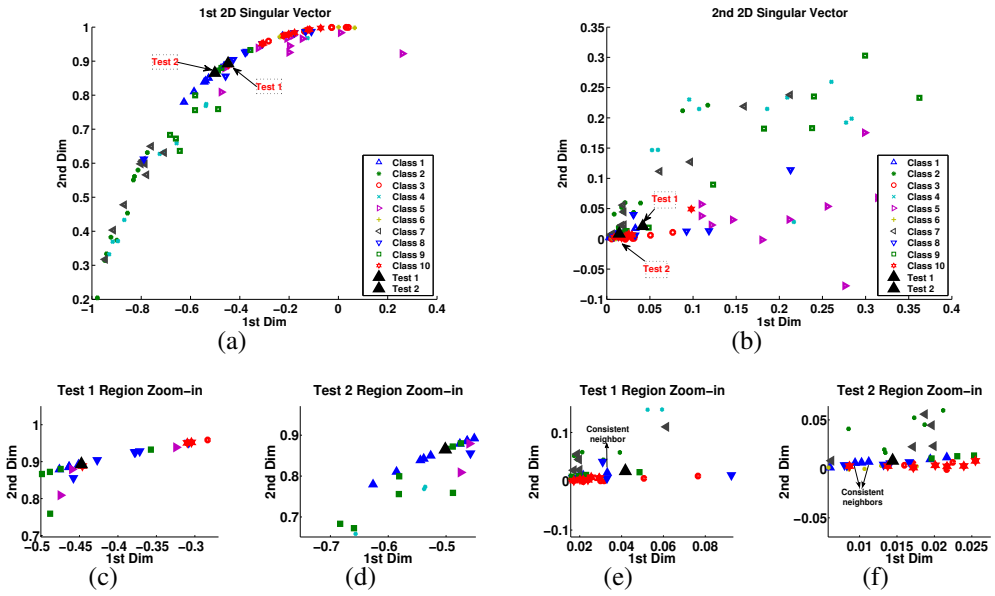

Figure B.13: (a) and (b) are 100 2D training sub-features (Eq.(3)) obtained from Australian Sign Language (Auslan) database by performing SVD over each training MTS regarding only the x,y attributes; (c)-(f) are region zoom-in of test point 1 (test 1) and test point 2 (test 2) in the 1st and 2nd 2D sub-feature space.

The robustness of the proposed CovSVDK feature extraction is illustrated in Fig. B.13 and Fig. B.14. Here Australian Sign Language (Auslan) database is employed by only keeping x,y information, such that each MTS contains two attributes, *i.e.,* $n = 2$. Applying SVD to each MTS yields two singular vectors, each with dimension 2. Fig. 13(a) and Fig. 13(b) are 2D sub-features (*i.e.,* $\frac{\lambda_1}{\|\rho\|_2}\mathbf{u}_1^T$ and $\frac{\lambda_2}{\|\rho\|_2}\mathbf{u}_2^T$) extracted from 100 training MTS data corresponding to the 1st and the 2nd singular vector respectively. Two test samples (Test 1 and Test 2), all associated to class 1, are selected. For Test 1, the same-class neighbor is consistently near it in both sub-feature spaces (Fig. 13(c) and Fig. 13(e)). For Test 2, three same-class neighbors are consistently close to it in both sub-feature spaces (Fig. 13(d) and Fig. 13(f)), while the similarities between Test 2 and samples from other classes vary significantly throughout the two sub-feature spaces. Using CovSVDK features, SRC can leverage the consistent closeness between the test sample and its same-class neighbors across different sub-feature spaces by computing a universal sparse code. Thus, Test 1 and Test 2 are correctly classified into class 1, as shown in Fig. 14(e) and Fig. 14(f). On the other hand, the classification scheme using one single singular vector, *i.e.,* Eq. (2), causes misclassification, as shown in Fig. 14(a)- 14(d).

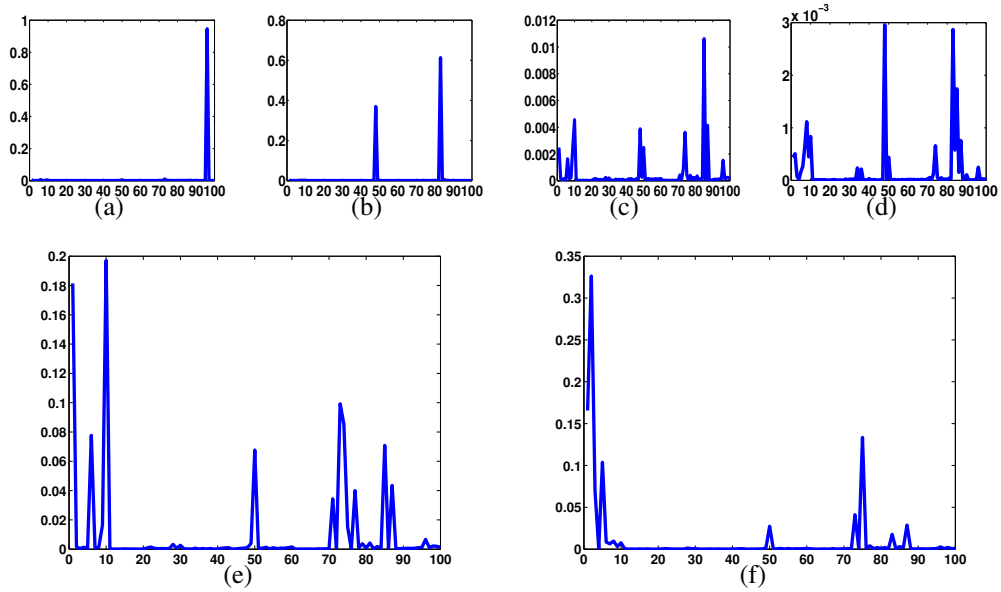

Figure B.14: Sparse codes computed based on different strategies. (a)-(d) are sparse codes solved based on Eq.(2) for each test point over the corresponding dictionary in each 2D feature space. (e) and (f) are sparse codes solved based on Eq.(10).