Information Sciences 292 (2015) 15–26

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Kernel sparse representation for time series classification

Zhihua Chen [a], Wangmeng Zuo [b,*], Qinghua Hu [c], Liang Lin [d]

[a] School of Software, Harbin Institute of Technology, Harbin, China
[b] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[c] School of Computer Science and Technology, Tianjin University, Tianjin, China
[d] School of Advanced Computing, Sun Yat-Sen University, Guangzhou, China

ARTICLE INFO

Article history:
Received 12 February 2014
Received in revised form 13 August 2014
Accepted 29 August 2014
Available online 8 September 2014

Keywords:
Sparse representation
Dictionary learning
Time series classification
Kernel method

ABSTRACT

In recent years there has been growing interests in mining time series data. To overcome the adverse influence of time shift, a number of effective elastic matching approaches such as dynamic time warp (DTW), edit distance with real penalty (ERP), and time warp edit distance (TWED) have been developed based on the nearest neighbor classification (NNC) framework, where the distance $d(\boldsymbol{x}, C_i)$ between a test sample $\mathbf{x}$ and one specific class C_i is simply defined as the minimum distance between $\mathbf{x}$ and the training samples in this class. In many applications, the sparse representation classifier (SRC) was applied by defining $d(\boldsymbol{x}, C_i)$ as the distance of $\boldsymbol{x}$ to a linear combination of the samples in class C_i, and it usually outperformed NNC in terms of classification accuracy. However, due to time shift, a linear combination of several time series is generally meaningless and may result in poor classification performance. In this paper, a family of Gaussian elastic matching kernels was introduced to deal with the problems of time shift and nonlinear representation. In this way, a linear combination of time series can be conducted in the implicit kernel space. Then a kernel sparse representation learning framework for time series classification was proposed. To improve computational efficiency and classification performance, both unsupervised and supervised dictionary learning techniques were developed by extending KSVD and label consistent KSVD algorithms. Experimental results showed that the proposed methods generally outperformed state-of-the-arts methods in terms of classification accuracy.

1. Introduction

A time series is a sequence of numerical values, typically measured at successive time instants spaced at uniform time intervals. It can be used to describe the states of objects and reflect their variations along time tags. For example, an electrocardiogram, which is widely used to analyze abnormal heart rhythms, is presented as a line graph, where the x-axis is time and the y-axis stands for the average voltage measured by the electrodes. Time series are used in various applications and can be easily acquired by the existing techniques. Clustering, classification, and mining of time series [9,24,30,38] have been extensively studied in many applications, such as sign language recognition [27], trajectory-based activity recognition [2], electrocardiography (ECG) based medical diagnosis [37], stock market time series categorization, and prediction [13,39].

* Corresponding author. Tel.: +86 451 86412871; fax: +86 451 86413309.
E-mail address: wmzuo@hit.edu.cn (W. Zuo).

http://dx.doi.org/10.1016/j.ins.2014.08.066
0020-0255/© 2014 Elsevier Inc. All rights reserved.

Similarity matching between time series is essential for time series classification [41]. There are two key problems in computing the distance or similarity between time series: time warping and high dimensionality. Time warping is a general phenomenon in time series, which creates huge challenges for automatic time series classification. For example, although time series can be viewed as points in vector space, conventional Euclidean distance is very sensitive to time warping and may fail in measuring similarity between time series. Dynamic time warping (DTW) was introduced to overcome the limitation of Euclidean distance [3,48]. However, DTW is time consuming. To improve the efficiency of DTW, Keogh and Ratanamahatana [21] proposed a lower bounding measure which significantly speeds up the calculation of DTW. To avoid the excessive distortion of DTW, Vlachos et al. [40] suggested a constraint of the longest common subsequence (LCSS) by assigning weights to different points. Chen et al. [7] introduced an edit distance on real sequence (EDR), which is robust against the factors of noise, time warping, and scaling. Unfortunately, DTW, LCSS, and EDR are not distance metrics as they do not satisfy the triangle inequality. Recently, Chen et al. [6] introduced a method called edit distance with real penalty (ERP) and Marteau [34] proposed an alignment-based distance metric, called time warp edit distance (TWED). Both of them satisfy the triangle inequality and are effective in measuring the dissimilarity between time series.

Using the existing distance measures, the nearest neighbor classification (NNC) framework is usually used for time series classification. Let the vector $\boldsymbol{y}$ be the test sample and the sample matrix $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K]$ be the training data matrix, where $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,ni}]$ is the sample matrix of class C_i, and $\boldsymbol{x}_{i,j}$ denotes the jth training sample of class C_i. Given a distance measure $d(\boldsymbol{x}, \boldsymbol{y})$, NNC [11] defines the distance of $\boldsymbol{y}$ to class C_i, $d(\boldsymbol{y}, C_i)$, as the minimum distance of $\{d(\boldsymbol{y}, \boldsymbol{x}_{ij}) \mid j = 1, \ldots, n_i\}$, and then $\boldsymbol{y}$ belongs to the class corresponding to the minimum of $\{d(\boldsymbol{y}, C_i)| \; i = 1, \ldots, K\}$, making the distance measure critical for NN-based classification and clustering [11,21,28].

Despite its popularity, the high dimensionality of samples and the limited size of training sets degrade the performance of NNC for time series classification. In recent years, a class of sparse representation based classifiers (SRC) [42,51] and collaborative representation based classifiers (CRC) [52] have been developed. SRC and CRC can achieve promising performance in many applications, such as face recognition [42,52], image classification [14], and traffic sign recognition [29]. SRC and CRC first assign a coefficient $.\alpha_{ij}$ for each training sample $\boldsymbol{x}_{ij}$, and then define the distance of $\boldsymbol{y}$ to the ith class C_i as $d(\boldsymbol{y}, C_i) = \left\| \mathbf{y} - \sum_j \boldsymbol{x}_{ij}\alpha_{ij} \right\|$, where $\| \cdot \|$ is some vector norm. When the size of the training set is limited, SRC and CRC can jointly utilize all the training samples from class C_i to compute $d(\boldsymbol{y}, C_i)$, and usually achieve higher classification accuracy than conventional NNC methods. Zhang et al. [52] provided some geometric explanations of the working mechanism of SRC/CRC. To further enhance the discriminative ability and computational efficiency, researchers studied the dictionary learning problem by learning a set of atoms from the training set in both the unsupervised [1,46] and the supervised [14,32,31,54,45,19] manner.

In the field of time series classification, however, little attention has been given to the SRC/CRC based approaches. One possible reason may be that SRC/CRC operate in linear space while $d(\boldsymbol{y}, C_i)$ is computed based on the Euclidean distance. But for time series classification, Euclidean distance is sensitive to time warping and may achieve poor classification performance. Moreover, dictionary learning of time series is another challenging task. Recently, kernel SRC [15,53,47] and kernel dictionary learning [35,36] approaches have been proposed, which makes it possible to overcome the difficulties of applying SRC to time series classification.

In this paper, the application of sparse representation based classifiers to time series classification was investigated. First, by introducing a family of Gaussian elastic matching kernels, time series were embedded into an implicit reproducing kernel Hilbert space, which allowed the use of kernel SRC for time series classification. Second, based on Gaussian elastic matching kernels, the kernel KSVD algorithm for unsupervised dictionary learning of time series was used, making kernel SRC computationally efficient and scalable. Third, by incorporating class label information, a kernel version of the label consistent KSVD method (Kernel LC-KSVD) was proposed to further improve the discriminative capability of the learned dictionary. Finally, experimental results showed that, compared with NNC, significant improvement can be obtained by using the proposed kernel sparse representation based approaches. Moreover, the proposed kernel LC-KSVD method is more efficient because of the enhancement of the computational efficiency.

The remainder of the paper is organized as follows. In Section 2, a brief survey of the related work such as SRC and KSVD is provided. In Section 3, several Gaussian elastic matching kernels are introduced, and then kernel SRC is used for time series classification. In Section 4, the kernel KSVD algorithm is used for unsupervised dictionary learning, and then a kernel LC-KSVD method for supervised dictionary learning is proposed. In Section 4, the experimental results of the proposed methods are presented. Finally, Section 5 gives some concluding remarks.

2. Related work

2.1. Sparse representation-based classification

Denote $\boldsymbol{y}$ as the test sample and $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K]$ as the training data matrix, where $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,ni}]$ is the sample matrix of class C_i, $\boldsymbol{x}_{i,j}$ is the jth training sample from class C_i, and n_i stands for the total number of training samples of class C_i. Motivated by the compressed sensing theory [4,5,10], Wright et al. [42] proposed a sparse representation based classifier (SRC), where the test sample $\boldsymbol{y}$ is approximated by a linear combination of training samples from all classes with the l_1-norm sparsity regularization:

$$(\hat{\boldsymbol{\alpha}}) = \arg\ \min_{\boldsymbol{\alpha}}\left\{\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1\right\}, \tag{1}$$

where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1; \ldots, \boldsymbol{\alpha}_i; \ldots; \boldsymbol{\alpha}_K]$ and $\boldsymbol{\alpha}_i$ is the coding vector associated with $\boldsymbol{X}_i$. With the resolved $\hat{\boldsymbol{\alpha}}$, SRC assigns the class label of $\boldsymbol{y}$ by

$$Label(\boldsymbol{y}) = \arg\ \min_i\{d(\boldsymbol{y}, C_i) = \|y - \boldsymbol{X}_i\hat{\boldsymbol{\alpha}}_i\|_2\}. \tag{2}$$

Wright et al. used second-order cone programming or LASSO to solve this l_1-minimization problem [42]. Recently, Yang et al. [44] conducted a comparative study on several l_1-minimization solvers: the augmented Lagrangian method (ALM), interior-point method, fast gradient based methods, and homotopy. In addition to the l_1-norm regularizer, other forms of sparsity regularization [52,18,43,49] such as l_2-norm [52] and group sparsity [18] were studied for sparse representation based classification.

To capture the nonlinear similarity between samples, the kernel sparse representation method was developed [15,53,47]. Given the nonlinear mapping function $\Phi(\boldsymbol{x})$, the inner product in the implicit reproducing kernel Hilbert space (RKHS) can be defined by

$$K(\boldsymbol{x}, \boldsymbol{y}) = \langle\Phi(\boldsymbol{x}), \Phi(\boldsymbol{y})\rangle. \tag{3}$$

Let $\boldsymbol{\Phi}_i = [\Phi(\boldsymbol{x}_{i,1}), \Phi(\boldsymbol{x}_{i,2}), \ldots, \Phi(\boldsymbol{x}_{i,ni})]$ and $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \ldots, \boldsymbol{\Phi}_K]$. Kernel SRC was proposed to solve the following l_1-minimization problem:

$$(\hat{\boldsymbol{\alpha}}) = \arg\ \min_{\boldsymbol{\alpha}}\left\{\|\Phi(\boldsymbol{y}) - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1\right\}. \tag{4}$$

By introducing the kernel matrix $\boldsymbol{K} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$ and kernel vector $\boldsymbol{k} = \boldsymbol{\Phi}^T\Phi(y)$, the problem in Eq. (4) becomes

$$(\hat{\boldsymbol{\alpha}}) = \arg\ \min_{\boldsymbol{\alpha}}\left\{\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha} + 2\boldsymbol{k}^T\boldsymbol{\alpha} + \lambda\|\boldsymbol{\alpha}\|_1\right\}, \tag{5}$$

which can then be solved by the l_1-minimization solvers.

2.2. Dictionary learning

SRC can be viewed as a sparse coding problem, where the dictionary is defined as the entire set of training samples such as $\boldsymbol{D} = \boldsymbol{X}$. However, if the number of the training samples is large, the time of sparse coding rapidly increases. To improve the scalability and discriminative ability of SRC, dictionary learning was investigated to seek an appropriate and concise dictionary for classification. Dictionary learning approaches can be grouped into two categories: unsupervised and supervised dictionary learning. In unsupervised dictionary learning, the target is to find a compact dictionary, where each sample can be sparsely coded by the dictionary. FOCUSS [25], MOD [12], and KSVD [1] are several representative unsupervised methods, where the dictionary $\boldsymbol{D}$ is obtained by solving an optimization problem such as:

$$(\boldsymbol{D}, \boldsymbol{A}) = \arg\ \min_{\boldsymbol{D},\boldsymbol{A}}\|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{A}\|_F^2 \text{ s.t } \forall i, \|\boldsymbol{a}_i\|_0 \leqslant T_0, \tag{6}$$

where $\boldsymbol{a}_i$ denotes the ith column of $\mathbf{A}$, $\|\cdot\|_0$ denotes the l_0-norm which counts the number of non-zero elements of a vector, and $\|\cdot\|_F$ denotes the Frobenius norm. Dictionary learning is a non-convex optimization problem, and most algorithms learn the dictionary $\boldsymbol{D}$ by iterating between updating $\mathbf{A}$ and updating $\boldsymbol{D}$.

Supervised dictionary learning aims to learn a discriminative dictionary from the training set $\boldsymbol{X}$ by incorporating the class label information. Several supervised dictionary learning approaches such as discriminative KSVD [54], task-driven dictionary learning [31], Fisher discrimination dictionary learning [45], and label-consistent KSVD (LC-KSVD) [19,20] have been proposed. The loss function of supervised dictionary learning generally includes both a reconstruction term and a discrimination term. For example, the loss function of LC-KSVD is defined as

$$(\boldsymbol{D}, \boldsymbol{W}, \boldsymbol{B}, \boldsymbol{A}) = \arg\min_{\boldsymbol{D},\boldsymbol{W},\boldsymbol{B},\boldsymbol{A}}\|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{A}\|_F^2 + \alpha\|\boldsymbol{Q} - \boldsymbol{B}\boldsymbol{A}\|_F^2 + \beta\|\boldsymbol{H} - \boldsymbol{W}\boldsymbol{A}\|_F^2 \text{ s.t } \forall i, \|\boldsymbol{a}_i\|_0 \leqslant T_0, \tag{7}$$

where $\boldsymbol{W}$ is the classification parameters, $\boldsymbol{A}$ is the discriminative sparse codes of the input samples $\boldsymbol{X}$, α and β control the relative contribution of the two discrimination terms, and the matrices $\boldsymbol{Q}$ and $\boldsymbol{H}$ are defined based on the class label information. The model in Eq. (7) can be reformulated as

$$(\boldsymbol{D}, \boldsymbol{W}, \boldsymbol{B}, \boldsymbol{A}) = \arg\ \min_{\boldsymbol{D},\boldsymbol{W},\boldsymbol{B},\boldsymbol{A}}\left\|\begin{pmatrix}\boldsymbol{X}\\ \sqrt{\alpha}\boldsymbol{Q}\\ \sqrt{\beta}\boldsymbol{H}\end{pmatrix} - \begin{pmatrix}\boldsymbol{D}\\ \sqrt{\alpha}\boldsymbol{B}\\ \sqrt{\beta}\boldsymbol{W}\end{pmatrix}\boldsymbol{A}\right\|_F^2 \text{ s.t } \forall i, \|\boldsymbol{a}_i\|_0 \leqslant T_0, \tag{8}$$

and can be efficiently solved using the existing K-SVD solver. Please refer to [19,20] for more details on LC-KSVD.

3. SRC with the Gaussian elastic matching kernel

Although time series can be simply represented as a 1D vector, the conventional SRC in Eq. (1) usually cannot achieve satisfactory classification performance due to the property of time warping and high dimensionality of the data. Motivated by the success of elastic matching methods, the Gaussian RBF kernel can be generalized into a class of Gaussian elastic matching kernel. Then, a SRC with Gaussian elastic matching kernel method for time series classification is suggested.

3.1. Gaussian elastic matching kernel

Given two samples $\boldsymbol{x}$ and $\boldsymbol{y}$, the Gaussian RBF kernel is defined as

$$K(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{\sigma^2}\right), \tag{9}$$

where σ is the hyper-parameter of the Gaussian RBF kernel, $\|\boldsymbol{x}-\boldsymbol{y}\|^2$ is the square of the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. Motivated by the success of elastic matching methods, a class of Gaussian elastic matching kernels was produced by substituting the Euclidean distance with the elastic distance measures such as DTW, ERP, and TWED.

3.1.1. Gaussian DTW kernel

Dynamic time warping (DTW) [3,21] is widely applied in time series classification and clustering. Given two time series $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]$ and $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]$, where $x_i(y_i)$ denotes the ith element of the time series $\boldsymbol{x}(\boldsymbol{y})$, the DTW distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is recursively defined as

$$d_{dtw}\left(\boldsymbol{x}_1^m,\boldsymbol{y}_1^n\right) = |x_m - y_n| + \min\begin{cases} d_{dtw}\left(\boldsymbol{x}_1^{m-1},\boldsymbol{y}_1^n\right) \\ d_{dtw}\left(\boldsymbol{x}_1^{m-1},\boldsymbol{y}_1^{n-1}\right), \\ d_{dtw}\left(\boldsymbol{x}_1^m,\boldsymbol{y}_1^{n-1}\right) \end{cases} \tag{10}$$

where $\boldsymbol{x}_p^q = (x_p, x_{p+1}, \ldots, x_q)$ denotes the subsequences of $\boldsymbol{x}$. By replacing the Euclidean distance in the Gaussian RBF kernel with the DTW distance, the Gaussian DTW kernel is defined as

$$K_{dtw}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{d_{dtw}(\boldsymbol{x},\boldsymbol{y})^2}{\sigma^2}\right). \tag{11}$$

The Gaussian DTW kernel had been studied in the support vector machine framework, but inconsistent classification performance was reported, partially because the Gaussian DTW kernel is not a positive semi-definite (PSD) kernel [38,16,17].

3.1.2. Gaussian ERP kernel

The edit distance with real penalty (ERP) [6] is a combination of the l_1 norm and the edit distance, which is a distance metric robust against time shifts. Given two time series $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]$ and $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]$, the ERP distance is recursively defined as,

$$d_{erp}\left(\boldsymbol{x}_1^m,\boldsymbol{y}_1^n\right) = \begin{cases} \sum_{i=1}^m |x_i - g|, & \text{if } n = 0 \\ \min\begin{cases} d_{erp}\left(\boldsymbol{x}_1^{m-1},\boldsymbol{y}_1^n\right) + |x_m - g| \\ d_{erp}\left(\boldsymbol{x}_1^{m-1},\boldsymbol{y}_1^{n-1}\right) + |x_m - y_n|, \\ d_{erp}\left(\boldsymbol{x}_1^m,\boldsymbol{y}_1^{n-1}\right) + |y_n - g| \end{cases} & \text{otherwise}, \\ \sum_{i=1}^n |y_i - g|, & \text{if } m = 0 \end{cases} \tag{12}$$

where g stands for the constant with the default value 0 [6]. Similarly, the Gaussian ERP kernel [50] is defined as

$$K_{erp}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{d_{erp}(\boldsymbol{x},\boldsymbol{y})^2}{\sigma^2}\right). \tag{13}$$

3.1.3. Gaussian TWED kernel

Marteau proposed the time warp edit distance (TWED) [34] by considering the time stamp factors of time series and applying the point pattern matching procedure [33] (PPM) to address time warping. TWED is also a distance metric. By taking into account time stamps, the time series are represented by $\boldsymbol{x} = [(x_1, t_{x1}), (x_2, t_{x2}), \ldots, (x_m, t_{xm})]$ and $\boldsymbol{y} = [(y_1, t_{y1}), (y_2, t_{y2}), \ldots, (y_n, t_{yn})]$, where $t_{xi}(t_{yi})$ stands for the time stamp of element $x_i(y_i)$. For any time series with $t_{xi} < t_{xj}$ and $\forall i < j$, the TWED distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is recursively defined as

$$d_{twed}\left(\boldsymbol{x}_1^m, \boldsymbol{y}_1^n\right) = \min \begin{cases} d_{twed}\left(\boldsymbol{x}_1^{m-1}, \boldsymbol{y}_1^n\right) + |x_m - x_{m-1}| + \gamma\left|t_{x_m} - t_{x_{m-1}}\right| + \lambda \\ d_{twed}\left(\boldsymbol{x}_1^{m-1}, \boldsymbol{y}_1^{n-1}\right) + |x_m - y_n| + \gamma\left|t_{x_m} - t_{y_n}\right| + |x_{m-1} - y_{n-1}| + \gamma\left|t_{x_{m-1}} - t_{y_{n-1}}\right|, \\ d_{twed}\left(\boldsymbol{x}_1^m, \boldsymbol{y}_1^{n-1}\right) + |y_n - y_{n-1}| + \gamma\left|t_{y_n} - t_{y_{n-1}}\right| + \lambda \end{cases} \tag{14}$$

where γ and λ are two non-negative constants. By substituting the Euclidean distance in the Gaussian RBF kernel with the TWED distance, the Gaussian TWED kernel [50] is defined as

$$K_{twed}(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{d_{twed}(\boldsymbol{x}, \boldsymbol{y})^2}{\sigma^2}\right). \tag{15}$$

3.2. Kernel SRC based time series classification

Denote $\boldsymbol{y}$ as the test sample and $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K]$ as the training data matrix, where $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,ni}]$, $i = 1, 2, \ldots, K$, is the sample matrix of class C_i, n_i stands for the number of training samples in class C_i, and $\boldsymbol{x}_{i,j}$ denotes the jth training time series of class C_i. The kernel function is defined as $K(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\mathbf{y})\rangle$, where $\Phi(\boldsymbol{x})$ stands for the corresponding nonlinear mapping function of $\boldsymbol{x}$. In kernel SRC, the dictionary is defined as $\mathbf{\Phi} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, \ldots, \mathbf{\Phi}_K]$, where $\mathbf{\Phi}_i = [\Phi(\boldsymbol{x}_{i,1}), \Phi(\boldsymbol{x}_{i,2}), \ldots, \Phi(\boldsymbol{x}_{i,ni})]$. Given a test sample $\boldsymbol{y}$, the following kernel sparse representation model for time series classification was used:

$$(\hat{\boldsymbol{\alpha}}) = \arg\min_{\alpha}\left\{\|\Phi(\boldsymbol{y}) - \mathbf{\Phi}\boldsymbol{\alpha}\|_2^2\right\}, \text{ s.t. } \|\boldsymbol{\alpha}\|_0 \leqslant T_0, \tag{16}$$

where $\hat{\boldsymbol{\alpha}} = [\hat{\boldsymbol{\alpha}}_1, \hat{\boldsymbol{\alpha}}_2, \ldots, \hat{\boldsymbol{\alpha}}_K]^T$ is the optimal solution. After obtaining $\hat{\boldsymbol{\alpha}}$, SRC assigns the class label of $\boldsymbol{y}$ by using the following rule

$$Label(\boldsymbol{y}) = \arg\ \min_i\left\{\|\Phi(\boldsymbol{y}) - \mathbf{\Phi}_i\hat{\boldsymbol{\alpha}}_i\|_2\right\}. \tag{17}$$

By introducing the kernel matrix $\boldsymbol{K} = \mathbf{\Phi}^T\mathbf{\Phi}$ and kernel vector $\boldsymbol{k} = \mathbf{\Phi}^T\Phi(\boldsymbol{y})$, the kernel orthogonal matching pursuit (OMP) algorithm [36] was modified to solve the kernel sparse representation model in Eq. (16). Let $\hat{\boldsymbol{\alpha}}_s$ be the current estimate of $\hat{\boldsymbol{\alpha}}$, and I_s be the set of indices of selected atoms. The residue $\boldsymbol{r}_s$ is defined as

$$\boldsymbol{r}_s = \Phi(y) - \mathbf{\Phi}\hat{\boldsymbol{\alpha}}_s. \tag{18}$$

The first step of kernel OMP is the projection of the residual to each of the remaining atoms,

$$\tau_i = \langle \boldsymbol{r}_s, \Phi(\boldsymbol{x}_i)\rangle = K(\boldsymbol{y}, \boldsymbol{x}_i) - \sum_{j \in I_s} K(\boldsymbol{x}_j, \boldsymbol{x}_i)\alpha_j, i \notin I_s. \tag{19}$$

Let

$$i_{\max} = \arg\max|\tau_i|. \tag{20}$$

The kernel OMP simply updates the set of indices $I_{s+1} = I_s \cup i_{max}$, and updates $\hat{\boldsymbol{\alpha}}_{s+1}$ by

$$\hat{\boldsymbol{\alpha}}_{s+1} = \boldsymbol{K}_{s+1}^{-1}\boldsymbol{k}_{s+1}, \tag{21}$$

where $\boldsymbol{K}_{s+1}$ is the sub-matrix of $\boldsymbol{K}$ based on the index set I_{s+1}, and $\boldsymbol{k}_{s+1}$ is the sub-vector of $\boldsymbol{k}$ based on the index set I_{s+1}. Finally, the kernel OMP algorithm for kernel sparse representation is summarized in Algorithm 1.

Given one test sample for the first class in the Face (Four) dataset, by applying the Gaussian ERP kernel, Fig. 1 shows the coding coefficients of kernel SRC on 24 training samples and the distance to each class. It can be seen that the coding coefficients corresponding to the first class are relatively significant, and the distance to the first class is 0.0577, which is much

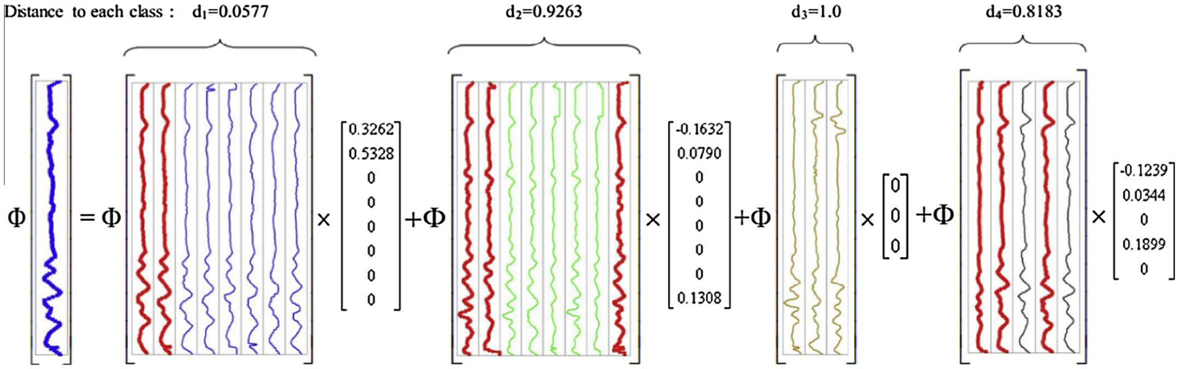

Fig. 1. Kernel sparse representation of time series.

smaller than those to the other classes. From Fig. 1, most coefficients of the other classes are zeros (compared to the first class, coefficients from other classes are much smaller), which indicates the effectiveness of kernel SRC.

Algorithm 1. Kernel OMP for kernel sparse representation

```
Input: time series $y$, training set $X$, $T_0$
Output: $\hat{\boldsymbol{\alpha}}$
Initialize $s = 0$, $I_0 = \emptyset$, $\hat{\boldsymbol{\alpha}}_0 = 0$
While $s < T_0$
    $\tau_i = \boldsymbol{K}(\boldsymbol{y}, \boldsymbol{x}_i) - \sum_{j \in I_s} K(\boldsymbol{x}_j, \boldsymbol{x}_i)\alpha_j, i \notin I_s$
    $i_{\max} = \arg\max|\tau_i|$
    Update the index set: $I_{s+1} = I_s \cup i_{max}$
    $\hat{\boldsymbol{\alpha}}_{s+1} = \boldsymbol{K}_{s+1}^{-1}\boldsymbol{k}_{s+1}$
    $s = s + 1$
End while
$\hat{\boldsymbol{\alpha}}(I_s(j)) = \hat{\boldsymbol{\alpha}}_s(j)$ for $\forall j \in I_s$, and otherwise zero.
```

3.3. Discussion

Because of time distortion, conventional sparse representation performed in the Euclidean space cannot work well for time series classification. Considering the success of elastic matching methods such as DTW, ERP, and TWED, a class of Gaussian elastic matching kernel was introduced. Based on the Gaussian elastic matching kernel, a kernel SRC model together with a kernel OMP algorithm was proposed. By substituting the Euclidean distance with the elastic distance measures, the Gaussian elastic matching kernel utilized kernel SRC to suppress the adverse influence of time drift.

It should be noted that the Gaussian elastic matching kernel cannot be guaranteed to be a positive definite symmetric (PDS) kernel. Empirical studies by Lei and Sun showed that the Gaussian DTW kernel is not PDS acceptable to support vector machine [26]. Experimental results showed that in some cases SVM with the Gaussian DTW kernel even performed poorer than SVM with the Gaussian RBF kernel or NNC with DTW. Fortunately, as shown in our previous studies [50], the Gaussian elastic matching kernel based on elastic metric (ERP or TWED) generally satisfied the PDS property. Moreover, even the Gaussian elastic matching kernel is not PDS, the proposed modification methods [8] can make the non-PDS kernel acceptable to kernel SRC. In practice, first whether $\boldsymbol{K}$ or $\boldsymbol{K}_s$ are PDS is checked. If $\boldsymbol{K}$ or $\boldsymbol{K}_s$ is not PDS, then non-PDS $\boldsymbol{K}$ or $\boldsymbol{K}_s$ is replaced with the proper PDS matrices by using the spectrum clip method [8].

4. Unsupervised and supervised dictionary learning of time series

Dictionary learning makes the SRC model in Eq. (16) applicable to a large scale training set, and enhances the discrimination of the dictionary. In this section, first a kernel KSVD with Gaussian elastic matching kernel for unsupervised dictionary learning is introduced, and then a kernel LC-KSVD method for supervised dictionary learning is proposed.

4.1. Kernel KSVD for unsupervised dictionary learning

In the sparse representation model in Eq. (16), the dictionary is simply the whole set of training samples. In this subsection, the kernel KSVD algorithm [35] is used to learn a more representative dictionary. Given the training set $\boldsymbol{X}$ of n samples, the goal of kernel KSVD is to learn a dictionary $\Phi(\boldsymbol{D})$ of m atoms by solving the following optimal problem,

$$(\boldsymbol{D}, \boldsymbol{A}) = \arg\ \min_{\boldsymbol{D},\boldsymbol{A}}\left\{\|\Phi(\boldsymbol{X}) - \Phi(\boldsymbol{D})\boldsymbol{A}\|_F^2\right\}, \quad \text{s.t. } \|\boldsymbol{a}_i\|_0 \leqslant T_0, \tag{22}$$

where $\boldsymbol{a}_i$ is the ith column of matrix $\boldsymbol{A}$. To make the model easy to solve, it is assumed that the dictionary atom is represented by a linear combination of the training samples such as $\Phi(\boldsymbol{D}) = \Phi(\boldsymbol{X})\boldsymbol{B}$, where $\boldsymbol{B}$ is the $n \times m$ atom representation dictionary. Thus, the kernel KSVD model is reformulated as

$$(\boldsymbol{B}, \boldsymbol{A}) = \arg\ \min_{\boldsymbol{D},\boldsymbol{A}}\left\{\|\Phi(\boldsymbol{X}) - \Phi(\boldsymbol{X})\boldsymbol{B}\boldsymbol{A}\|_F^2\right\}, \quad \text{s.t. } \|\boldsymbol{a}_i\|_0 \leqslant T_0. \tag{23}$$

The proposed kernel dictionary learning method [35] is used to learn the dictionary by iterating between the updating of the coding coefficients $\boldsymbol{A}$ and the updating of the dictionary $\boldsymbol{B}$.

4.1.1. Updating *A* via kernel OMP

Given the atom representation dictionary $\boldsymbol{B}$,$\boldsymbol{A}$ is updated by solving n independent sparse coding problems,

$$(\boldsymbol{a}_i) = \arg\ \min_{\boldsymbol{D},\boldsymbol{A}}\left\{\|\Phi(\boldsymbol{x}_i) - \Phi(X)\boldsymbol{B}\boldsymbol{a}_i\|_2^2\right\}, \quad \text{s.t. } \|\boldsymbol{a}_i\|_0 \leqslant T_0. \tag{24}$$

Let $\hat{\boldsymbol{\alpha}}_s$ be the current estimate of $\hat{\boldsymbol{\alpha}}$ and I_s be the set of indices of selected atoms. The residue $\boldsymbol{r}_s$ is defined as

$$\boldsymbol{r}_s = \Phi(\boldsymbol{y}) - \boldsymbol{\Phi B}\hat{\boldsymbol{\alpha}}_s. \tag{25}$$

The first step of KOMP is the projection of the residual to each of the remaining atoms,

$$\tau_i = \langle \boldsymbol{r}_s, \Phi(\boldsymbol{X})\boldsymbol{b}_i\rangle = \boldsymbol{k}^T\boldsymbol{b}_i - a_S^T\boldsymbol{B}^T\boldsymbol{K}\boldsymbol{b}_i,\ i \notin I_s. \tag{26}$$

Let

$$i_{\max} = \arg\max|\tau_i|. \tag{27}$$

The kernel OMP simply updates the set of indices $I_{s+1} = I_s \cup i_{max}$, and constructs the sub-matrix $\boldsymbol{K}_{s+1}$. If $\boldsymbol{K}_{s+1}$ is not semi-positive definite, a semi-positive definite approximation is used by solving the following problem,

$$\widetilde{\boldsymbol{K}}_{s+1} = \arg\ \min_{K\succ 0}\|\boldsymbol{K} - \boldsymbol{K}_{s+1}\|_F^2. \tag{28}$$

According to Chen et al. [8], $\widetilde{\boldsymbol{K}}_{s+1}$ can be easily obtained using the spectrum clipping operator. Based on $\widetilde{\boldsymbol{K}}_{s+1}, \hat{\boldsymbol{\alpha}}_{s+1}$ is updated by,

$$\hat{\boldsymbol{\alpha}}_{s+1} = \widetilde{\boldsymbol{K}}_{s+1}^{-1}\boldsymbol{k}_{s+1}, \tag{29}$$

where $\boldsymbol{k}_{s+1}$ is the sub-vector of $\boldsymbol{k}$ based on the index set I_{s+1}. This procedure is repeated until T_0 atoms are selected.

4.1.2. Dictionary updating

In the dictionary update phase, first the spectrum clipping operator on $\boldsymbol{K}$ is used to obtain a PDS approximation $\widetilde{\boldsymbol{K}}$. Denoting $\boldsymbol{b}_k$ by the kth column of $\boldsymbol{B}$ and $\boldsymbol{a}_j$ the jth row of $\boldsymbol{A}$, the error matrix $\boldsymbol{E}_k$ is computed as

$$\boldsymbol{E}_k = \left(\boldsymbol{I} - \sum_{j\neq k}\boldsymbol{b}_j\boldsymbol{a}_j\right). \tag{30}$$

Let ω_k be the group of examples that use the kth atom and $\boldsymbol{\Omega}_k$ be an $n \times |\omega_k|$ matrix with $\boldsymbol{\Omega}_k(\omega_k(i), i) = 1$ and zero elsewhere. The column reduced error matrix is then obtained by $\boldsymbol{E}_k^R = \boldsymbol{E}_k\boldsymbol{\Omega}_k$. By applying SVD to $(\boldsymbol{E}_k^R)^T\widetilde{\boldsymbol{K}}\boldsymbol{E}_k^R$,

$$\left(\boldsymbol{E}_k^R\right)^T\widetilde{\boldsymbol{K}}\boldsymbol{E}_k^R = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T, \tag{31}$$

$\boldsymbol{a}_k$ is updated by

$$\boldsymbol{a}_k = (\lambda_1)^{-1/2}\boldsymbol{E}_k^R\boldsymbol{v}_1, \tag{32}$$

where λ_1 is the first eigenvalue and $\boldsymbol{v}_1$ is the first eigenvector. This procedure is repeated until all the m atoms are updated.

4.2. Kernel label consistent K-SVD for supervised dictionary learning

To further enforce the compactness and discrimination of the dictionary, a kernel label consistent KSVD (LC-KSVD) algorithm is proposed. Using the Gaussian elastic matching kernel, the loss function of kernel LC-KSVD is defined as

$$< \boldsymbol{D}, \boldsymbol{W}, \boldsymbol{U}, \boldsymbol{A} >= \arg\min_{\boldsymbol{D},\boldsymbol{W},\boldsymbol{U},\boldsymbol{A}}\|\Phi(\boldsymbol{X}) - \Phi(\boldsymbol{D})\boldsymbol{A}\|_2^2 + \alpha\|\boldsymbol{Q} - \boldsymbol{UA}\|_2^2 + \beta\|\boldsymbol{H} - \boldsymbol{WA}\|_2^2, \|\boldsymbol{a}_i\|_0 \leqslant T_0, \tag{33}$$

The first term is the standard dictionary learning model. In the second term, $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n]$, n is the number of training samples in $\boldsymbol{X}$, and $\boldsymbol{q}_i$ is a discriminative sparse code corresponding to the ith training sample. If the training sample $\boldsymbol{x}_i$ shares the same label with dictionary atom $\Phi(\boldsymbol{d}_j)$, the jth element of $\boldsymbol{q}_i$ will be one and otherwise zero. In the third item, $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n]$, where $\boldsymbol{h}_i$ is a label vector corresponding to $\boldsymbol{x}_i$. If $\boldsymbol{x}_i$ belongs to the kth class, the kth element of $\boldsymbol{h}_i$ will be one and otherwise zero. Also, α and β are two non-negative parameters, and $\boldsymbol{U}$ and $\boldsymbol{W}$ are two transformation matrices to be learned.

The kernel LC-KSVD model can be equivalently formulated as

$$< \boldsymbol{D}, \boldsymbol{W}, \boldsymbol{U}, \boldsymbol{A} >= \arg\min_{\boldsymbol{D},\boldsymbol{W},\boldsymbol{U},\boldsymbol{A}}\left\|\begin{pmatrix}\Phi(\boldsymbol{X})\\ \sqrt{\alpha}\boldsymbol{Q}\\ \sqrt{\beta}\boldsymbol{H}\end{pmatrix} - \begin{pmatrix}\Phi(\boldsymbol{D})\\ \sqrt{\alpha}\boldsymbol{U}\\ \sqrt{\beta}\boldsymbol{W}\end{pmatrix}\boldsymbol{A}\right\|, \|\boldsymbol{a}_i\|_0 \leqslant T_0. \tag{34}$$

By introducing an implicit mapping

$$\Psi\begin{pmatrix}\boldsymbol{x}_i\\ \sqrt{\alpha}\boldsymbol{q}_i\\ \sqrt{\beta}\boldsymbol{h}_i\end{pmatrix} = \begin{pmatrix}\Phi(\boldsymbol{x}_i)\\ \sqrt{\alpha}\boldsymbol{q}_i\\ \sqrt{\beta}\boldsymbol{h}_i\end{pmatrix}, \tag{35}$$

the corresponding kernel function is defined as

$$K'\left(\begin{pmatrix}\boldsymbol{x}_i\\ \sqrt{\alpha}\boldsymbol{q}_i\\ \sqrt{\beta}\boldsymbol{h}_i\end{pmatrix},\begin{pmatrix}\boldsymbol{x}_j\\ \sqrt{\alpha}\boldsymbol{q}_j\\ \sqrt{\beta}\boldsymbol{h}_j\end{pmatrix}\right)=\left\langle \Psi\begin{pmatrix}\boldsymbol{x}_i\\ \sqrt{\alpha}\boldsymbol{q}_i\\ \sqrt{\beta}\boldsymbol{h}_i\end{pmatrix},\Psi\begin{pmatrix}\boldsymbol{x}_j\\ \sqrt{\alpha}\boldsymbol{q}_j\\ \sqrt{\beta}\boldsymbol{h}_j\end{pmatrix}\right\rangle=K(\boldsymbol{x}_i,\boldsymbol{x}_j)+\alpha\langle\boldsymbol{q}_i,\boldsymbol{q}_j\rangle+\beta\langle\boldsymbol{h}_i,\boldsymbol{h}_j\rangle. \tag{36}$$

Let $\mathbf{z}_i=[x_i^T,\sqrt{\alpha}\boldsymbol{q}_i^T,\sqrt{\beta}\boldsymbol{h}_i^T]^T$. The kernel LC-KSVD model can then be formulated as,

$$<\boldsymbol{B},\boldsymbol{A}>=\arg\min_{\boldsymbol{B},\boldsymbol{A}}\|\Phi'(\boldsymbol{Z})-\Phi'(\boldsymbol{Z})\boldsymbol{BA}\|_F^2 \quad \text{s.t. } \|\boldsymbol{a}_i\|_0\leqslant T,\forall i, \tag{37}$$

which can be solved by using the kernel KSVD algorithm introduced in Section 4.1. Once $\boldsymbol{B}$ is obtained, $\boldsymbol{U}$ and $\boldsymbol{W}$ can be acquired via $\boldsymbol{U}=\boldsymbol{QB},\boldsymbol{W}=\boldsymbol{HB}$, respectively.

In the classification stage, given the test sample $\boldsymbol{y}$, first the sparse representation problem is solved by

$$(\hat{\boldsymbol{a}})=\arg\ \min_{\boldsymbol{a}}\left\{\|\Phi(\boldsymbol{y})-\Phi(\boldsymbol{X})\boldsymbol{Ba}\|_2^2\right\},\quad \text{s.t. } \|\boldsymbol{a}\|_0\leqslant T_0, \tag{38}$$

and then $\boldsymbol{y}$ is classified based on the following rule:

$$j=\arg\ \max_j\{\boldsymbol{l}=\boldsymbol{W}\hat{\boldsymbol{a}}\}, \tag{39}$$

where $\boldsymbol{l}$ is the $K\times 1$ class label vector.

5. Experimental results

In this section, a series of experiments were conducted to assess the proposed methods using the UCR time series datasets [22,23] from two aspects: classification error rate and computational cost. Sixteen data sets were used in the experiments, four of which were two-class tasks and the rest were multi-class problems. Each dataset consists of a training subset and a test subset. Table 1 provides a brief summary of the datasets.

5.1. Classification results and analysis

Using the classification error rate as the performance indicator, the proposed methods were compared with the state-of-the-art algorithms based on the nearest neighbor classifier, including NNC with Euclidean (1NN-ED), NNC with DTW (1NN-DTW), NNC with ERP (1NN-ERP), and NNC with TWED (1NN-TWED). The proposed methods were grouped into three categories. The first one was based on the kernel SRC model where the dictionary is the entire set of the training samples, and three kernel SRC methods, SRC with Gaussian DTW kernel (SRC-DTW), SRC with Gaussian ERP kernel (SRC-ERP), and SRC with TWED kernel (SRC-TWED) were evaluated. The second category was based on kernel KSVD, where the dictionary is learned using the kernel KSVD algorithm in the unsupervised way, and three KSVD approaches, kernel KSVD with Gaussian DTW kernel (KSVD-DTW), kernel KSVD with Gaussian ERP kernel (KSVD-ERP), and kernel KSVD with Gaussian TWED kernel (KSVD-TWED) were evaluated. The third category was based on the kernel label consistent KSVD algorithm, where the dictionary is learned in the supervised manner, and three kernel LC-KSVD approaches, LC-KSVD with Gaussian DTW kernel (LC-KSVD-DTW), LC-KSVD with Gaussian ERP kernel (LC-KSVD-ERP), and LC-KSVD with Gaussian TWED kernel (LC-KSVD-TWED)

Table 1
Attributes of the UCR time series datasets.

Datasets	Class	Length	Instances	
			Training	Test
Adiac	37	176	390	391
Beef	5	470	30	30
CBF	3	128	30	900
Coffee	2	286	28	28
ECG200	2	96	100	100
Face(All)	14	131	560	1690
Face(Four)	4	350	24	88
FISH	7	463	175	175
Gun-point	2	150	50	150
Lighting2	2	637	60	61
Lighting7	7	319	70	73
Olive oil	4	570	30	30
Swedish leaf	15	128	500	625
Synthetic control	6	60	300	300
Trace	4	275	100	100
Two patterns	4	128	1000	4000

Table 2
Classification error rates obtained by the conventional 1NN classifier with different distance measures and the sparse representation methods with different Gaussian elastic kernels.

	1NN-ED	1NN-DTW	1NN-ERP	1NN-TWED	SRC-DTW	SRC-ERP	SRC-TWED	KSVD-DTW	KSVD-ERP	KSVD-TWED	LC-KSVD -DTW	LC-KSVD -ERP	LC-KSVD-TWED
Adiac	0.389	0.396	0.378	0.376	0.412	0.376	0.373	0.412	0.332	0.284	0.427	0.402	0.409
Beef	0.467	0.500	0.500	0.533	0.433	0.400	0.333	0.300	0.233	0.300	0.300	0.300	0.300
CBF	0.148	0.003	0.003	0.007	0	0.003	0	0	0.004	0	0	0.002	0.002
Coffee	0.250	0.179	0.250	0.214	0.036	0.036	0	0.036	0	0	0	0.036	0
ECG200	0.120	0.230	0.130	0.100	0.170	0.130	0.080	0.180	0.120	0.070	0.150	0.100	0.080
Face(All)	0.286	0.192	0.202	0.189	0.240	0.196	0.191	0.217	0.164	0.127	0.225	0.199	0.205
Face(Four)	0.216	0.170	0.102	0.034	0.125	0.045	0.034	0.114	0.080	0.034	0.148	0.080	0.034
FISH	0.217	0.167	0.120	0.057	0.131	0.120	0.057	0.189	0.126	0.057	0.171	0.103	0.057
Gun-point	0.087	0.093	0.040	0.013	0.090	0.033	0.007	0.073	0.007	0.007	0.073	0.007	0.013
Lighting2	0.246	0.131	0.148	0.131	0.164	0.164	0.131	0.230	0.180	0.131	0.180	0.115	0.115
Lighting7	0.425	0.274	0.301	0.247	0.233	0.301	0.205	0.205	0.178	0.178	0.205	0.192	0.192
Olive oil	0.133	0.133	0.167	0.167	0.167	0.133	0.133	0.133	0.133	0.100	0.133	0.133	0.100
Swedish leaf	0.213	0.210	0.120	0.104	0.210	0.100	0.099	0.150	0.075	0.090	0.194	0.101	0.094
Synthetic_control	0.120	0.007	0.036	0.023	0.013	0.023	0.007	0.010	0.007	0.007	0.013	0.010	0.010
Trace	0.240	0	0.170	0.050	0.010	0.120	0	0.010	0.070	0.010	0	0.110	0.020
Two patterns	0.090	0	0	0	0	0	0	0	0	0	0	0	0

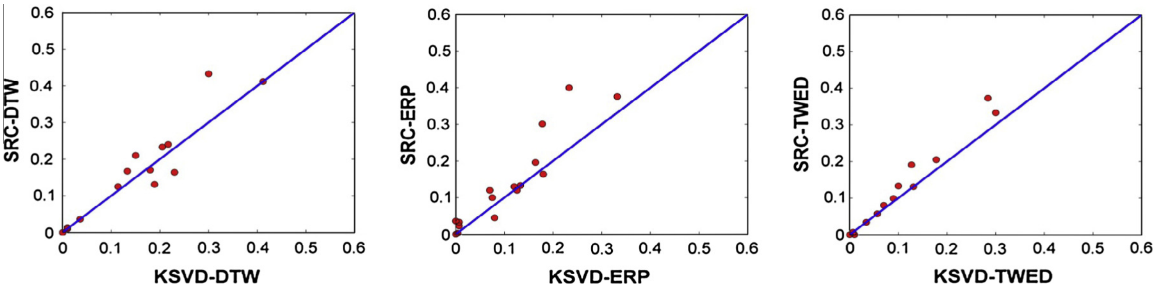

Fig. 2. Comparison between the kernel KSVD model and the kernel SRC model. In each sub-graph, the *x* axis stands for the error rates generated by the kernel KSVD model with a certain Gaussian kernel (DTW, ERP, TWED respectively), and the *y* axis represents the error rates formed by the kernel SRC model with the same kernel shown on the *x*axis. The straight line has a slope of 1.0, so a dot on the line means the identical error rate calculated by the two methods on the same data set, a dot above (or below) the line means the KSVD model performs better (or weaker) than the SRC model on that data set.

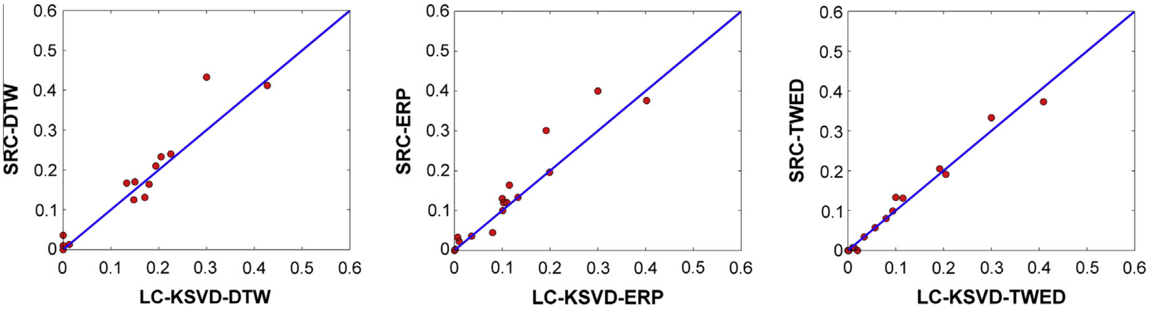

Fig. 3. Comparison between the kernel LC-KSVD model and the kernel SRC model. In each sub-graph, the *x* axis stands for the error rates generated by the kernel LC-KSVD model with a certain Gaussian kernel (DTW, ERP, TWED respectively), the *y* axis represents the error rates formed by the kernel SRC model with the same kernel shown on the *x*axis. The straight line has a slope of 1.0, so a dot on the line means the identical error rate calculated by the two methods on the same data set, a dot above (or below) the line means the LC-KSVD model performs better (or weaker) than the SRC model on that data set.

were evaluated. The codes of the proposed methods are available at: https://github.com/voidman2009/Kernel-Sparse-Representation/tree/master/KernelKSVD%20for%20time%20series.

Table 2 lists the classification error rates of all the methods on these 16 data sets. Generally, the proposed sparse representation based classification methods were superior to the state-of-the-art nearest neighbor classifiers. For example, SRC-ERP can achieve lower error rates than 1NN-ERP on 14 data sets. Only on the CBF dataset, 1NN-ERP was 0.001 lower than SRC-ERP. For all the 16 data sets, KSVD-TWED always obtained a lower or equal error rate than 1NN-TWED. It is

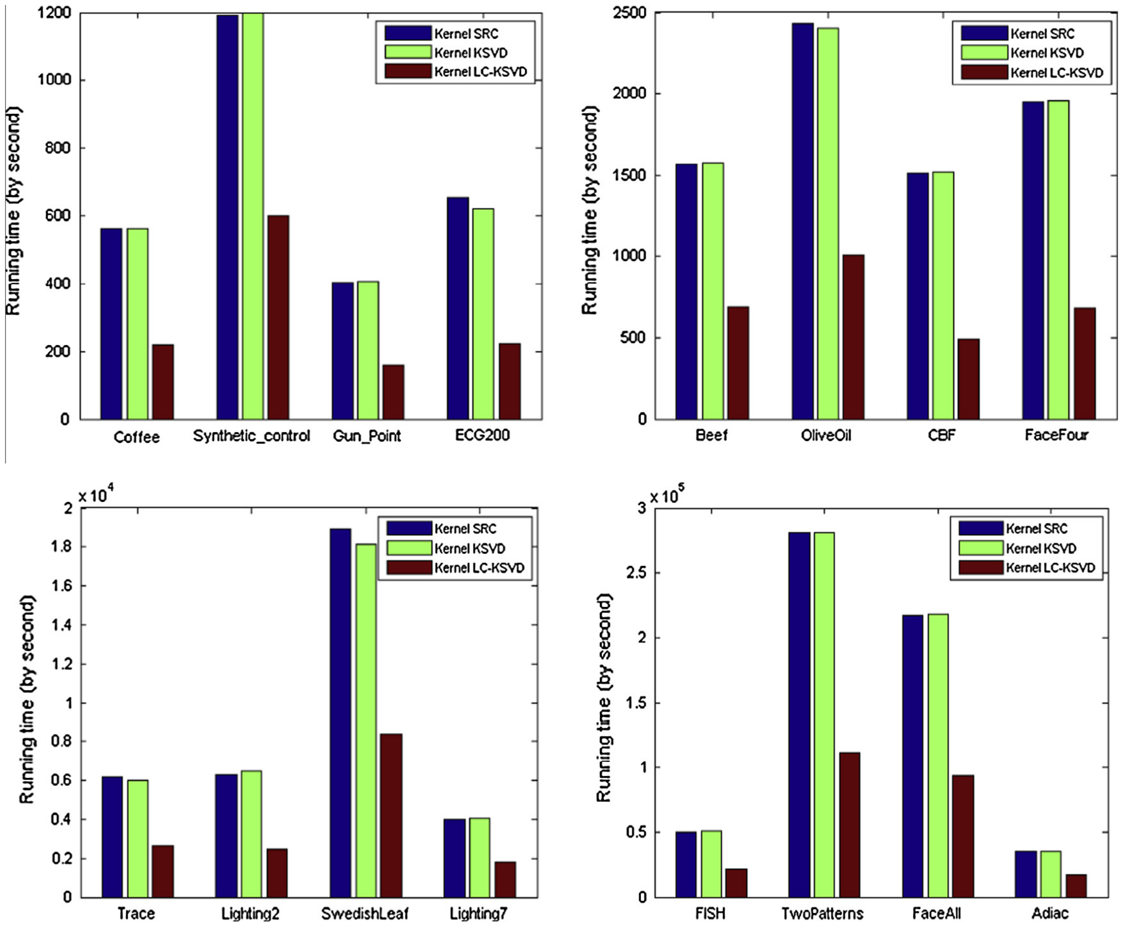

Fig. 4. Running time of kernel SRC, kernel KSVD, and kernel LC-KSVD on the 16 data sets (the Gaussian TWED kernel is used as an example).

interesting to point out that compared with ERP and TWED, the improvement of SRC-DTW against 1NN-DTW is relatively weak, which may be explained in that ERP and TWED are distance metrics while DTW is not.

In addition, based on the experimental results, kernel LC-KSVD and kernel KSVD can to some extent achieve better performance than kernel SRC. For example, LC-KSVD-ERP and KSVD-ERP achieved lower classification accuracy than SRC-ERP only on 3 and 4 data sets, respectively. KSVD-TWED showed weakness only on *Trace* by 0.01 lower than SRC-TWED, but the results generated by LC-KSVD-TWED and SRC-TWED were roughly the same. In order to give a clearer comparison, the performances of these 3 categories of methods were analyzed.

From Fig. 2, it can be seen that KSVD-DTW performs a little better than SRC-DTW. KSVD-ERP is more effectively than SRC-ERP. The KSVD-TWED outperforms SRC-TWED because there is hardly any dot below the line. From Fig. 3, for the Gaussian DTW kernel and the Gaussian ERP kernel, the kernel LC-KSVD model is superior to the kernel SRC model. However, LC-KSVD-TWED and SRC-TWED (the right sub-graph) present similar performance since most of the dots in this sub-graph are close to the straight line.

5.2. Running time

In this section, using the Gaussian TWED kernel, the running time of kernel SRC, kernel KSVD, and kernel LC-KSVD were compared. Fig. 4 shows the running time of these three methods on the 16 data sets. In the classification stage, the procedures of kernel SRC and kernel KSVD are similar. If the size of the dictionary of kernel SRC is the same as that of kernel KSVD, the computational cost of these two methods would be roughly the same. In the training stage, the number of atoms was set be the same as the number of training samples. From Fig. 4, it can be seen that the difference of running time of kernel SRC and kernel KSVD is insignificant. In kernel LC-KSVD, the number of atoms was set much lower than the size of the training set, and the classification rule was simpler than those of kernel SRC and kernel KSVD. Thus, the running time of kernel LC-KSVD was much less than the other methods. Taking both the classification accuracy and running time into account, kernel LC-KSVD is a suitable choice for time series classification.

6. Conclusion

In this paper, the applications of kernel sparse representation based classifiers for time series classification were studied. The introduction of a class of Gaussian elastic matching kernels, the Gaussian DTW kernel, the Gaussian ERP kernel, and the Gaussian TWED kernel, makes it possible to utilize SRC while suppressing the influence of time drift. Both the kernel sparse representation and dictionary learning methods were investigated, and three kernel sparse representation based classifiers, including kernel SRC, kernel KSVD, and kernel LC-KSVD were proposed. Experimental results on the UCR time series datasets showed that the proposed methods can achieve much lower error rates than the state-of-the-art nearest neighbor classifiers, 1NN-DTW, 1NN-ERP, and 1NN-TWED. Moreover, the running time of kernel LC-KSVD was much less than kernel SRC and kernel KSVD. In the future, we will study the construction of elastic PDS kernels and will develop more appropriate discriminative dictionary learning algorithms for time series classification.

Acknowledgement

This work was supported by National Program on Key Basic Research Project (973 Program) under Grant 2013CB329304, National Natural Science Foundation of China (NSFC) under Grants 61271093, 61222210, 61170107, 61471146, and New Century Excellent Talents in University under Grant NCET-12-0150 and NCET-12-0399. Thanks to Dr. Edward C. Mignot, Shandong University, for linguistic advice.

References

[1] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.

[2] F.I. Bashir, A.A. Khokhar, D. Schonfeld, Object trajectory-based activity classification and recognition using hidden Markov models, IEEE Trans. Image Process. 16 (7) (2007) 1912–1919.

[3] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, AAAI-94 Workshop on Knowledge Discovery in Databases, 1994, pp. 229–248.

[4] E. Candès, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Commun. Pure Appl. Math. 59 (8) (2006) 1207–1223.

[5] E. Candès, T. Tao, Decoding by linear programming, IEEE Trans. Inform. Theory 51 (12) (2005) 4203–4215.

[6] L. Chen, R. Ng, On the marriage of Lp-norms and edit distance, in: Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB'04), 2004, pp. 792–803.

[7] L. Chen, M.T. Özsu,V. Oria, Robust and fast similarity search for moving object trajectories, in: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD'05), 2005, pp. 491–502.

[8] Y.H. Chen, E.K. Garcia, M.R. Gupta, A. Rahimi, L. Cazzanti, Similarity-based classification: concepts and algorithms, J. Machine Learn. Res. 10 (2009) 747–776.

[9] H. Deng, G. Runger, E. Tuv, M. Vladimir, A time series forest for classification and feature extraction, Inform. Sci. 239 (2013) 142–153.

[10] D.L. Donoho, Compressed sensing, IEEE Trans. Inform. Theory 52 (4) (2006) 1289–1306.

[11] R.O. Duta, P.E. Hart, D.G. Stork, Pattern Classification, second ed., John Wiley & Sons, New York, 2000.

[12] K. Engan, S.O. Aase, J.H. Husøy, Multi-frame compression: theory and design, EURASIP Signal Process. 80 (10) (2000) 2121–2140.

[13] T.C. Fu, C.W. Law, K.K. Chan, F.L. Chung, C.M. Ng, Stock time series categorization and clustering via SB-tree optimization, in: Proceedings of the Third International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'06), 2006, pp. 1130–1139.

[14] S. Gao, I.W.-H. Tsang, Y. Ma, Learning category-specific dictionary and shared dictionary for fine-grained image categorization, IEEE Trans. Image Process. 23 (2) (2014) 623–634.

[15] S. Gao, I.W. Tsang, L.T. Chia, Sparse representation with kernels, IEEE Trans. Image Process. 22 (2) (2013) 423–434.

[16] S. Gudmundsson, T.P. Runarsson, S. Sigurdsson, Support vector machines and dynamic time warping for time series, in: IEEE International Joint Conference on Neural Networks (IJCNN'08), 2008, pp. 2772–2776.

[17] A. Jalalian, S.K. Chalup, GDTW-P-SVMs: variable-length time series analysis using support vector machines, Neurocomputing 99 (1) (2013) 270–282.

[18] K. Jia, T.H. Chan, Y. Ma, Robust and practical face recognition via structured sparsity, in: Proceedings of the 12th European Conference on Computer Vision (ECCV'12), 2012, pp. 331–344.

[19] Z.L. Jiang, Z. Lin, L.S. Davis, Learning a discriminative dictionary for sparse coding via label consistent K-SVD, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), 2011, pp. 1697–1704.

[20] Z. Jiang, Z. Lin, L.S. Davis, Label consistent K-SVD: learning adiscriminative dictionary for recognition, IEEE Trans. Pattern Anal. Machine Intell. (2013).

[21] E. Keogh, C.A. Ratanamahatana, Exact indexing of dynamic time warping, Knowl. Inform. Syst. 7 (3) (2005) 358–386.

[22] E. Keogh, S. Kasetty, On the need for time series data mining benchmarks: a survey and empirical demonstration, Data Mining Knowl. Discovery 7 (4) (2003) 349–371.

[23] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, C.A. Ratanamahatana, The UCR Time Series Classification/Clustering, 2011. <www.cs.ucr.edu/∼eamonn/time_series_data/>.

[24] M. Krawczak, G. Szkatuła, An approach to dimensionality reduction in time series, Inform. Sci. 260 (2014) 15–36.

[25] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, T.J. Sejnowski, Dictionary learning algorithms for sparse representation, Neural Comput. 15 (2) (2003) 349–396.

[26] H.S. Lei, B.Y. Sun, A study on the dynamic time warping in kernel machines, in: Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (SITIS '07), 2007, pp. 839–845.

[27] J.F. Lichtenauer, E.A. Hendriks, M.J.T. Reinders, Sign language recognition by combining statistical DTW and independent classification, IEEE Trans. Pattern Anal. Machine Intell. 30 (11) (2008) 2040–2046.

[28] L. Lin, X. Liu, S.-C. Zhu, Layered graph matching with composite cluster sampling, IEEE Trans. Pattern Anal. Machine Intell. 32 (8) (2010) 1426–1442.

[29] H. Liu, Y. Liu, F. Sun, Traffic sign recognition using group sparse coding, Inform. Sci. 266 (2014) 75–89.

[30] E.A. Maharaj, P. D'Urso, Fuzzy clustering of time series in the frequency domain, Inform. Sci. 181 (7) (2011) 1187–1211.

[31] J. Mairal, F. Bach, J. Ponce, Task-driven dictionary learning, IEEE Trans. Pattern Anal. Machine Intell. 34 (4) (2012) 791–804.

[32] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Supervised dictionary learning, Adv. Neural Inform. Process. Syst. (2008).

[33] V. Mäkinen, Using edit distance in point-pattern matching, in: Proc. 8th International Symposium on String Processing and Information Retrieval (SPIRE'01), 2001, pp. 153–161.

[34] P.F. Marteau, Time warp edit distance with stiffness adjustment for time series matching, IEEE Trans. Pattern Anal. Machine Intell. 31 (2) (2009) 306–318.

[35] H.V. Nguyen, V.M. Patel, N.M. Nasrabadi, R. Chellappa, Kernel dictionary learning, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'12), 2012, pp. 2021–2024.

[36] H. Van Nguyen, V.M. Patel, N.M. Nasrabadi, R. Chellappa, Design of non-linear kernel dictionaries for object recognition, IEEE Trans. Image Process. 22 (12) (2013) 5123–5135.

[37] K. Noponen, J. Kortelainen, T. Seppänen, Invariant trajectory classification of dynamical systems with a case study on ECG, Pattern Recogn. 42 (9) (2009) 1832–1844.

[38] H. Pree, B. Herwig, T. Gruber, B. Sick, K. David, P. Lukowicz, On general purpose time series similarity measures and their use as kernel functions in support vector machines, Inform. Sci. 281 (2014) 478–495.

[39] M. Pulido, P. Melin, O. Castillo, Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange, Inform. Sci. 280 (2014) 188–204.

[40] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, E. Keogh, Indexing multi-dimensional time-series with support for multiple distance measures, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), 2003, pp. 216–225.

[41] X.Y. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, E. Keogh, Experimental comparison of representation methods and distance measures for time series data, Data Mining Knowl. Discovery 26 (2013) 275–309.

[42] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Machine Intell. 31 (2) (2009) 210–227.

[43] Y. Xu, Q. Zhu, Z. Fan, D. Zhang, J. Mi, Z. Lai, Using the idea of the sparse representation to perform coarse-to-fine face recognition, Inform. Sci. 238 (2013) 138–148.

[44] A.Y. Yang, Z.H. Zhou, A.G. Balasubramanian, S. Sastry, Y. Ma, Fast l1-minimization algorithms for robust face recognition, IEEE Trans. Image Process. 22 (8) (2013) 3234–3246.

[45] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher discrimination dictionary learning for sparse representation, in: 2011 IEEE International Conference on Computer Vision (ICCV'11), 2011, pp. 543–550.

[46] S. Yang, Y. Lv, Y. Ren, L. Yang, L. Jiao, Unsupervised images segmentation via incremental dictionary learning based sparse representation, Inform. Sci. 269 (2014) 48–59.

[47] J. Yin, Z.H. Liu, Z. Jin, W.K. Yang, Kernel sparse representation based classification, Neurocomputing 77 (1) (2012) 120–128.

[48] D. Yu, X. Yu, Q. Hu, J. Liu, A. Wu, Dynamic time warping constraint learning for large margin nearest neighbor, Inform. Sci. 181 (13) (2011) 2787–2796.

[49] B. Zhang, F. Karray, Q. Li, L. Zhang, Sparse representation classifier for microaneurysm detection and retinal blood vessel extraction, Inform. Sci. 200 (2012) 78–90.

[50] D. Zhang, W.M. Zuo, D. Zhang, H.Z. Zhang, Time series classification using support vector machine with Gaussian elastic metric kernel, in: Proceedings of 20th International Conference on Pattern Recognition (ICPR'10), 2010, pp. 29–32.

[51] H.C. Zhang, N.M. Nasrabadi, Y.N. Zhang, T.S. Huang, Joint dynamic sparse representation for multi-view face recognition, Pattern Recogn. 45 (4) (2012) 1290–1298.

[52] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition, in: 2011 IEEE International Conference on Computer Vision (ICCV'11), 2011, pp. 471–478.

[53] L. Zhang, W.D. Zhou, P.C. Chang, J. Liu, Z. Yan, T. Wang, F.Z. Li, Kernel sparse representation-based classifier, IEEE Trans. Signal Process. 60 (4) (2012) 1684–1695.

[54] Q. Zhang, B.X. Li, Discriminative K-SVD for dictionary learning in face recognition, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), 2010, pp. 2691–2698.