Neurocomputing 77 (2012) 120–128

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

# Kernel sparse representation based classification

Jun Yin [a,*], Zhonghua Liu [a], Zhong Jin [a], Wankou Yang [b]

[a] School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China
[b] School of Automation, Southeast University, Nanjing 210096, China

ARTICLE INFO

*Article history:*
Received 22 December 2010
Received in revised form
23 August 2011
Accepted 28 August 2011
Communicated by Y. Fu
Available online 22 September 2011

*Keywords:*
Classification
Sparse representation
Kernel

ABSTRACT

Sparse representation has attracted great attention in the past few years. Sparse representation based classification (SRC) algorithm was developed and successfully used for classification. In this paper, a kernel sparse representation based classification (KSRC) algorithm is proposed. Samples are mapped into a high dimensional feature space first and then SRC is performed in this new feature space by utilizing kernel trick. Since samples in the high dimensional feature space are unknown, we cannot perform KSRC directly. In order to overcome this difficulty, we give the method to solve the problem of sparse representation in the high dimensional feature space. If an appropriate kernel is selected, in the high dimensional feature space, a test sample is probably represented as the linear combination of training samples of the same class more accurately. Therefore, KSRC has more powerful classification ability than SRC. Experiments of face recognition, palmprint recognition and finger-knuckle-print recognition demonstrate the effectiveness of KSRC.



## 1. Introduction

In pattern recognition, classification is an indispensable step and classifier design is one of the most popular technologies. Several classification approaches have been proposed over the past several decades [1,2]. Among them, the nearest-neighbor (NN) classifier and the nearest-mean (NM) classifier are most widely used because of their simpleness and availability. The NN classifier assigns a test sample to the category of its nearest neighbor from the labeled training set. Instead of searching the nearest training sample, the NM classifier assigns a test sample to the category of its nearest class mean.

Over the last more than 10 years, the kernel based algorithms [3] such as kernel principal component analysis (KPCA) [4] and kernel fisher discriminant analysis (KFD) [5,6] have aroused considerable interest in pattern recognition and machine learning. These kernel based algorithms improve the computational power of the linear algorithms. They map the data into a high dimensional feature space by a nonlinear mapping and perform linear algorithms in the high dimensional feature space using the inner products. In the high dimensional feature space, the inner products can be computed by a kernel function. For classification, utilizing kernel approach, Yu et al. present the kernel nearest neighbor (KERNEL-NN) classifier [7]. KERNEL-NN applies the nearest neighbor classification method in the high dimensional feature space. Kernel approach could change the distribution of samples by the nonlinear mapping. Some linearly inseparable samples in the original feature space can become linearly separable in the high dimensional feature space. If an appropriate kernel is chosen to reshape the distribution of samples, the KERNEL-NN classifier could perform better than the NN classifier.

Recently, sparse representation becomes a hot topic of pattern recognition and computer vision. It is applied to image super-resolution [8], motion segmentation [9] and supervised denoising [10]. Wright et al. apply sparse representation to classification and exploit the sparse representation based classification (SRC) algorithm [11]. For SRC, a test sample is represented as a sparse combination of training samples, and its sparse representation coefficient is obtained by solving the problem of sparse representation. The test sample is assigned to the class that minimizes the residual between itself and the reconstruction constructed by training samples of this class. SRC shows its effectiveness in face recognition experiments.

As well known, after mapping samples into a high dimensional feature space by a nonlinear mapping, kernel approach can change the distribution of samples. If an appropriate kernel function is utilized, for a test sample, more neighbors probably have the same class label as itself in the high dimensional feature space. Here, in the high dimensional feature space, the test sample can be represented more accurately as the combination of training samples from the same class. Then the nonzero entries of sparse representation coefficient vector of the test sample will be more associated with training samples from the same class as itself. Namely, sparse representation coefficient in the high dimensional

* Corresponding author.
*E-mail address:* yinjun8429@163.com (J. Yin).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.08.018

feature space can denote the category of the test sample more accurately and it has more powerful discriminating ability. To use sparse representation coefficient in the high dimensional feature space, we propose the kernel sparse representation based classification (KSRC) algorithm in this paper. For KSRC, samples are mapped into a high dimensional feature space first and then SRC is performed in this new feature space. We prove that SRC in the high dimensional feature space can be formulated in terms of the inner products, while the inner products could be computed by kernel function. Comprehensive comparisons between KSRC and NM, NN, KERNEL-NN and SRC reveal the superior characteristics of KSRC.

The rest of the paper is organized as follows: Section 2 describes SRC algorithm and proposes KSRC algorithm. Section 3 describes experiments on several popular databases. Finally, the conclusions are summarized in Section 4.

## 2. The proposed algorithm

### 2.1. Sparse representation based classification (SRC) [11]

Suppose that we have $n$ training samples for $c$ classes and sufficient training samples belong to the $k$th class, $A_k=[x_{k,1}, x_{k,2},\ldots,x_{k,n_k}]\in\mathbb{R}^{m\times n_k}$, where $m$ is the dimension of samples and $n_k$ is the number of training samples of the $k$th class. Any test sample $y\in\mathbb{R}^m$ from the $k$th class can be approximately represented as the linear combination of training samples of this class:

$$y=\alpha_{k,1}x_{k,1}+\alpha_{k,2}x_{k,2}+\cdots+\alpha_{k,n_k}x_{k,n_k} \tag{1}$$

Since the label of $y$ is unknown initially, we represent $y$ as the linear combination of all the training samples:

$$y=A\alpha_0 \tag{2}$$

where $A=[A_1,A_2,\ldots,A_c]=[x_{1,1},x_{1,2},\ldots,x_{c,n_c}]\in\mathbb{R}^{m\times n}$ is a matrix composed of all the $n$ training samples of $c$ classes and $\alpha_0=[0,\ldots,0,\alpha_{k,1},\alpha_{k,2},\ldots,\alpha_{k,n_k},0,\ldots,0]^T\in\mathbb{R}^n$ is the coefficient vector whose nonzero entries are only associated with the $k$th class. When $c$ is large, $\alpha_0$ will be sparse.

If $m<n$, Eq. (2) is underdetermined. The problem of sparse representation is to search a vector $\alpha$ such that Eq. (2) is satisfied and $\|\alpha\|_0$ is minimized, where $\|\alpha\|_0$ is the $l_0$-norm of $\alpha$. This can be described as

$$\hat{\alpha}_0=\arg\min_{\alpha}\|\alpha\|_0 \text{ subject to } y=A\alpha \tag{3}$$

However, finding the sparse solution of Eq.(3) is NP-hard [12]: namely, there is no known procedure for obtaining the sparsest solution, which is significantly more efficient than exhausting all subsets of the entries for $\alpha$. The theory of sparse representation and compressive sensing [13–15] reveals that we can solve the following convex relaxed optimization to obtain approximate solution:

$$\hat{\alpha}_1=\arg\min_{\alpha}\|\alpha\|_1 \text{ subject to } y=A\alpha \tag{4}$$

where $\|\alpha\|_1$ is the $l_1$-norm of $\alpha$. This problem can be solved by standard linear programming methods [16]. Furthermore, the observations are often inaccurate, then we should relax the constraint in Eq. (4) and get the following optimization problem:

$$\hat{\alpha}_1=\arg\min_{\alpha}\|\alpha\|_1 \text{ subjectto } \|A\alpha-y\|_2\le\varepsilon \tag{5}$$

where $\varepsilon$ is the tolerance of the reconstruction error. This convex optimization problem can be solved via second-order cone programming [16].

The optimization problem (5) is mainly used to deal with small noise. In practice, the observations possibly contain big noise. For example, the images are corrupted or occluded. Here, the errors cannot be ignored or solved by the optimization problem (5). The constraint should be modified as

$$y=A\alpha+e=[A\ I]\begin{bmatrix}\alpha\\ e\end{bmatrix} \tag{6}$$

where $e\in\mathbb{R}^m$ is a vector of errors, $I\in\mathbb{R}^{m\times m}$ is the identity matrix. Now, we get the following optimization problem:

$$\hat{\gamma}_1=\arg\min_{\gamma}\|\gamma\|_1 \text{ subject to } y=P\gamma \tag{7}$$

where

$$P=[A\ \ I]\in\mathbb{R}^{m\times(n+m)},\gamma=\begin{bmatrix}\alpha\\ e\end{bmatrix}\in\mathbb{R}^{n+m} \quad\text{and}\quad \hat{\gamma}_1=\begin{bmatrix}\hat{\alpha}_1\\ \hat{e}_1\end{bmatrix}\in\mathbb{R}^{n+m}$$

Let $\hat{\alpha}_1$ denote the solution of sparse representation problem (7) obtained by $l_1$-minimization. Ideally, the nonzero entries in $\hat{\alpha}_1$ will be associated with the columns of $A$ from a single object class, and we can easily assign the test sample $y$ to that class. However, noise and modeling error may cause small nonzero entries associated with multiple classes. Simple heuristics such as assigning $y$ to the class with the largest entry are not dependable. Instead, we define a new vector $\hat{\alpha}_1^k(k=1,2,\ldots,c)$ whose only nonzero entries are the entries in $\hat{\alpha}_1$ that are associated with class $k$. The reconstruction with the training samples of the $k$th class is $\hat{y}_k=A\hat{\alpha}_1^k(k=1,2,\ldots,c)$. Then $y$ can be assigned to the class that minimizes the residual between $y$ and $\hat{y}_k$:

$$\min_k r_k(y)=\|y-A\hat{\alpha}_1^k\|_2 \tag{8}$$

The SRC algorithm is summarized as follows:

**Algorithm 1.** Sparse representation based classification (SRC)

1. Input: the matrix of training samples $A\in\mathbb{R}^{m\times n}$, a test sample $y\in\mathbb{R}^m$.
2. Normalize the columns of $A$ to have unit $l_2$-norm.
3. Solve the $l_1$-minimization problem defined in Eq. (4) or (5) or (7).
4. Compute the residuals $r_k(y)$ $(k=1,2,\ldots,c)$ defined in Eq. (8).
5. Output : $\text{identity}(y)=\arg\min_k(r_k(y))$

### 2.2. Kernel sparse representation based classification (KSRC)

As we know, kernel approach can change the distribution of samples by mapping samples into a high dimensional feature space [7]. This change possibly has two effects if an appropriate kernel function is selected. On the one hand, some linear inseparable samples in the original feature space become linear separable in the high dimensional feature space. This leads to superiority of the KERNEL-NN classifier over the NN classifier. On the other hand, a test sample can be represented as the linear combination of training samples from the same class as itself more accurately in the high dimensional feature space than original. Then the nonzero entries of sparse representation coefficient vector of the test sample are more associated with training samples of the same class. This results in better classification ability of SRC. So we perform SRC in the high dimensional feature space and propose kernel sparse representation based classification (KSRC). Because the explicit mapping from the original feature space to the high dimensional feature space is unknown, KSRC cannot be performed directly. However, we successfully solve this problem by Theorem 1.

Suppose that samples are mapped from original feature space $\mathbb{R}^m$ into a high dimensional feature space $H$ by a nonlinear

mapping $\phi$:

$$\mathbb{R}^m \to H, \quad x \to \phi(x) \tag{9}$$

Let $B=[\phi(x_{1,1}),\phi(x_{1,2}),\ldots,\phi(x_{c,n_c})]$ represent the matrix composed of all the training samples after the nonlinear mapping $\phi$. The problem of sparse representation in $H$ can be described as [11]

$$\hat{\beta}_0 = \arg\min_{\beta} \|\beta\|_0 \text{ subject to } \phi(y)=B\beta \tag{10}$$

where $\phi(y)$ is any test sample in the high dimensional feature space, which corresponds to $y$ in the original feature space. Similarly, the approximate solution of Eq. (10) can be obtained through the following convex relaxed optimization [13–15]:

$$\hat{\beta}_1 = \arg\min_{\beta} \|\beta\|_1 \text{ subject to } \phi(y)=B\beta \tag{11}$$

When the observations are not accurate, the constraint in Eq. (11) should be relaxed and the following optimization problem is obtained:

$$\hat{\beta}_1 = \arg\min_{\beta} \|\beta\|_1 \text{ subject to } \|B\beta-\phi(y)\|_2 \le \varepsilon \tag{12}$$

If we set $\varepsilon=0$, Eq. (12) is equivalent to Eq. (11). So Eq. (11) can be seen as a special case of Eq. (12) and we can only consider the optimization problem (12).

Since $B$ and $\phi(y)$ are unknown, Eq. (12) cannot be solved directly. But according to Theorem 1, Eq. (12) can be transformed to

$$\hat{\beta}_1 = \arg\min_{\beta} \|\beta\|_1 \text{ subject to } \|B^TB\beta-B^T\phi(y)\|_2 \le \delta \tag{13}$$

**Theorem 1.** *For any $\varepsilon \ge 0$, there must exist $\delta \ge 0$ such that we have $\|B\beta-\phi(y)\|_2 \le \varepsilon$, as long as $\|B^TB\beta-B^T\phi(y)\|_2 \le \delta$ is satisfied.*

The inner product of samples in the high dimensional feature space can be computed by kernel function. Namely, for any samples $x$ and $y$, we have $\phi(x)^T\phi(y)=k(x,y)$, where $k(x,y)$ is a kernel function. Then

$$B^TB = [\phi(x_{1,1}),\phi(x_{1,2}),\ldots,\phi(x_{c,n_c})]^T[\phi(x_{1,1}),\phi(x_{1,2}),\ldots,\phi(x_{c,n_c})] = \begin{bmatrix} k(x_{1,1},x_{1,1}) & k(x_{1,1},x_{1,2}) & \cdots & k(x_{1,1},x_{c,n_c}) \\ k(x_{1,2},x_{1,1}) & k(x_{1,2},x_{1,2}) & \cdots & k(x_{1,2},x_{c,n_c}) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_{c,n_c},x_{1,1}) & k(x_{c,n_c},x_{1,2}) & \cdots & k(x_{c,n_c},x_{c,n_c}) \end{bmatrix} \tag{14}$$

and

$$B^T\phi(y) = [\phi(x_{1,1}),\phi(x_{1,2}),\ldots,\phi(x_{c,n_c})]^T\phi(y) = \begin{bmatrix} k(x_{1,1},y) \\ k(x_{1,2},y) \\ \vdots \\ k(x_{c,n_c},y) \end{bmatrix} \tag{15}$$

When the kernel function $k(x,y)$ is given, $B^TB$ and $B^T\phi(y)$ are obtained. Now we could solve the convex optimization problem (13) via second-order cone programming [16]. If the observations contain big noise, as SRC, the constraint in Eq. (13) should be modified as

$$B^T\phi(y) = B^TB\beta+E = [B^TB\ \tilde{I}]\begin{bmatrix}\beta \\ E\end{bmatrix} \tag{16}$$

where $E \in \mathbb{R}^n$ is a vector of errors, $\tilde{I} \in \mathbb{R}^{n\times n}$ is the identity matrix. Utilizing constraint (16), the following optimization problem is obtained:

$$\hat{\eta}_1 = \arg\min_{\eta} \|\eta\|_1 \text{ subject to } B^T\phi(y) = Q\eta \tag{17}$$

where

$$Q=[B^TB\ \tilde{I}] \in \mathbb{R}^{n\times 2n}, \eta = \begin{bmatrix}\beta \\ E\end{bmatrix} \in \mathbb{R}^{2n} \quad \text{and} \quad \hat{\eta}_1 = \begin{bmatrix}\hat{\beta}_1 \\ \hat{E}_1\end{bmatrix} \in \mathbb{R}^{2n}$$

Let $\hat{\beta}_1$ denote the solution of optimization problem (17). Similar to SRC, we define a new vector $\hat{\beta}_1^k (k=1,2,\ldots,c)$ by setting only those entries in $\hat{\beta}_1$ associated with class $k$ nonzero and assigning zero to other entries. Then $y$ can be assigned to the class that minimizes the residual between $B^T\phi(y)$ and $B^TB\hat{\beta}_1^k$:

$$\min_k R_k(y) = \|B^T\phi(y)-B^TB\hat{\beta}_1^k\|_2 \tag{18}$$

**Algorithm 2. Kernel sparse representation based classification (KSRC)**

1. Input: the matrix of training samples $A \in \mathbb{R}^{m\times n}$, a test sample $y \in \mathbb{R}^m$ and a kernel function.
2. Normalize the columns of $A$ to have unit $l_2$-norm.
3. Calculate $B^TB$ and $B^T\phi(y)$ by Eqs. (14) and (15).
4. Solve the $l_1$-minimization problem defined in Eqs. (13) or (17).
5. Compute the residuals $R_k(y)$ $(k=1,2,\ldots,c)$ defined in Eq. (18).
6. Output : identity$(y) = \arg\min_k(R_k(y))$

For samples containing small noise, the computational cost of SRC and KSRC is mainly caused by solving the convex optimization problem (5) and (13), respectively. According to $A \in \mathbb{R}^{m\times n}$ and $B^TB \in \mathbb{R}^{n\times n}$, the computational complexity of solving Eqs. (5) and (13) are both $O(n^3)$. Here, SRC and KSRC have the same computational cost. For samples containing big noise, the computational cost of SRC and KSRC is mainly caused by solving the convex optimization problems (7) and (17) separately. We know that the size of $P$ is $m\times(n+m)$ and the size of $Q$ is $n\times 2n$. Then the computational complexity of solving Eq. (7) is $O((n+m)^3)$ and the computational complexity of solving Eq. (17) is $O((2n)^3)$ [23]. At this time, if the number of training sample size $n$ is smaller than the dimension $m$, the computational cost of KSRC is shorter than SRC. Otherwise, the computational cost of KSRC is longer than SRC.

## 3. Experiments

In this section, the effectiveness of KSRC algorithm is evaluated by experiments. We do experiments on FERET, ORL, Yale and AR face databases and the PolyU palmprint and finger-knuckle-print (FKP) databases. Principal component analysis (PCA) [17] and random projection (RP) [18] are used for feature extraction. We compare the classification ability of KSRC algorithm with NM, NN, KERNEL-NN and SRC algorithms after feature extraction. Two popular kernels are involved in our experiments. One is polynomial kernel $k(x,y)=(1+x^Ty)^d$ and the other is Gaussian kernel $k(x,y)=\exp(-\|x-y\|^2/t)$. For KSRC, we use these two kernels, respectively. Since KERNEL-NN [7] using Gaussian kernel is equivalent to NN, only polynomial kernel is used for KERNEL-NN. The optimal kernel parameters are selected.

### 3.1. Data corpora

#### 3.1.1. FERET face database

The FERET face database [19] was sponsored by the US Department of Defense through the DARPA Program. It has become a standard database for testing and evaluating face recognition algorithms. We perform algorithms on a subset of

the FERET face database. The subset is composed of 1400 images of 200 individuals, and each individual has seven images. It involves variations in face expression, pose and illumination. In the experiment, the facial portion of the original image was cropped based on the location of eyes and mouth. Then we resized the cropped images to $80 \times 80$ pixels and preprocess them by histogram equalization. Seven sample images of one person are shown in Fig. 1.

#### 3.1.2. ORL face database

ORL face database contains 400 face images of 40 individuals. The image size is $112 \times 92$ with 256 gray levels per pixel. The face images are centralized. There are variations in pose, illumination and facial expression. Fig. 2 shows sample images of one person.

#### 3.1.3. Yale face database

The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains 165 gray-scale images of 15 individuals. The images demonstrate variations in lighting, facial expression and with/without glasses. In our experiment, every image was manually cropped and resized to $100 \times 80$ pixels. Fig. 3 shows 11 images of one people.

#### 3.1.4. AR face database

The AR face database [20] contains over 4000 color face images of 126 people, including 26 frontal views of faces with different facial expressions, lighting conditions, and occlusions for each people. The pictures of 120 individuals were collected in two sessions (14 days apart) and each session contains 13 color images. Fourteen face images (each session containing 7) of these 120 individuals are selected in our experiment. The images are converted to grayscale. The face portion of each image is manually cropped and normalized to $50 \times 40$ pixels. Fig. 4 shows sample images of one person. These images vary as follows: neutral expression, smiling, angry, screaming, left light on, right light on, all sides light on.

#### 3.1.5. PolyU FKP database

FKP images on the PolyU FKP database were collected from 165 volunteers. These images are collected in two separate sessions. In each session, the subject was asked to provide six images for each of the left index finger, the left middle finger, the right index finger and the right middle finger. The images were processed by ROI extraction algorithm described in [21]. In the experiment, we select 1200 FKP images of the right index finger of 100 subjects. These selected images were resized to $55 \times 110$ pixels and preprocessed by histogram equalization. Fig. 5 shows 12 sample images of one right index finger.

#### 3.1.6. PolyU palmprint database

The PolyU palmprint database contains 600 images of 100 different palms with six samples for each palm. Six samples from each of these palms were collected in two sessions, where the first three were captured in the first session and the other three in

**Fig. 1.** Sample images of one person on FERET face database.

**Fig. 2.** Sample images of one person on ORL face database.

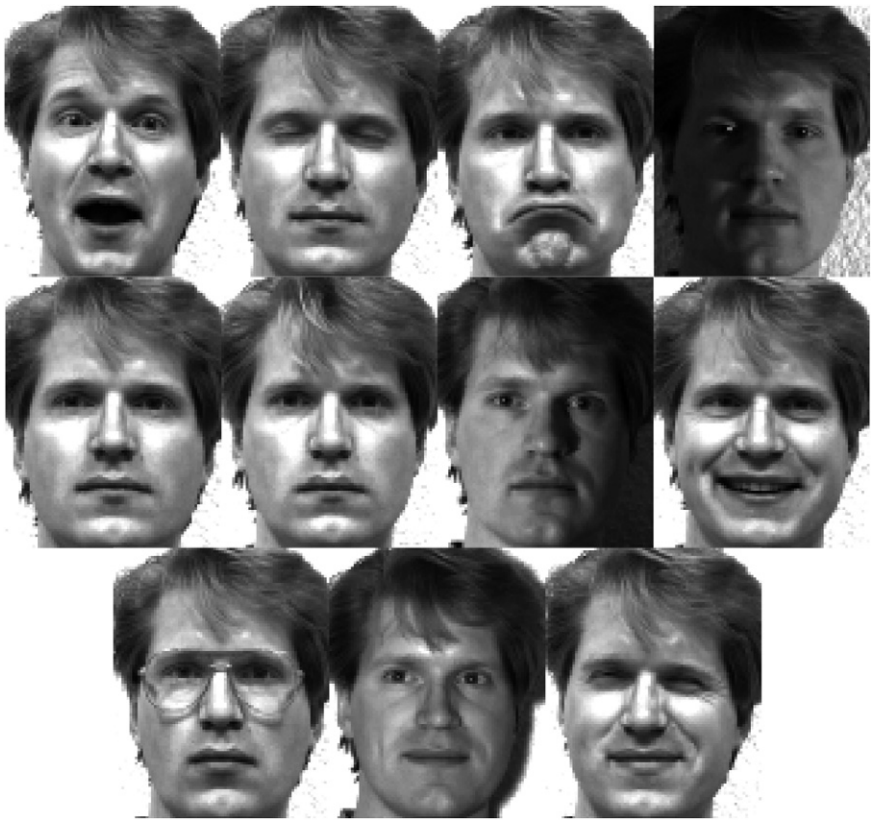

**Fig. 3.** Sample images of one person on Yale face database.

**Fig. 4.** Sample images of one person on AR face database.

the second session. The central part of each original image was automatically cropped using the algorithm mentioned in [22]. The cropped images were resized to $128 \times 128$ pixels and preprocessed by histogram equalization. Fig. 6 shows six sample images of one palm.

### 3.2. Experimental results

On FERET face database, first we try to find the optimal kernel parameters for KSRC using global-to-local search strategy [3]. Three images per person are randomly selected for training and the remaining four images are used for validation. After feature extraction by PCA, the dimension of the samples is fixed at 150. Through globally searching over a wide range of the parameter space, we find a candidate interval where the optimal parameters may exist. Here, for the parameter $d$ of polynomial kernel, the candidate interval is from 1 to 10, for the parameter $t$ of Gaussian kernel, the candidate interval is also from 1 to 10. Now, we try to find the optimal kernel parameters within these intervals. Fig. 7a shows the recognition rates of KSRC with polynomial kernel versus the variation of the parameter $d$. Fig. 7b shows the recognition rates of KSRC with Gaussian kernel versus the variation of the parameter $t$. From Fig. 7a and b, we can see that the optimal parameter $d$ is 2 and the optimal parameter $t$ is also 2. After determining the optimal kernel parameters, we compare KSRC with NM, NN, KERNEL-NN and SRC. The first three images per person are used for training and the rest four images are used for testing. Table 1 shows the maximal recognition rates of five methods and the corresponding dimensions and parameters. From Table 1, it can be seen that KSRC outperforms other four methods, whether polynomial kernel is used or Gaussian kernel is used.

In the first experiment on ORL face database, three images per individual are randomly chosen for training and the remaining seven images are used for testing. PCA is used for feature extraction. The experiment is repeated for 20 times. The first 10 times are used

for tuning kernel parameters and the other 10 times for comparing the performance of NM, NN, KERNEL-NN, SRC and KSRC. The optimal kernel parameters are also determined by global-to-local search strategy. Fig. 8 shows the average recognition rates versus the dimensions. Table 2 lists the maximal average recognition rate and the standard deviation of each method across 10 runs and the corresponding dimension and parameter. From Fig. 8 and Table 2, we can see four main points. First, no matter which kernel is used, our KSRC consistently outperforms other four algorithms irrespective of the variation of dimensions. Second, SRC performs better than NM, NN and KERNEL-NN algorithms when the dimension is over about 20. Third, KERNEL-NN almost outperforms NN. Last, NM has the worst performance in this experiment. From the first and the third points, we can see that kernel approach indeed improve the classification ability.

We know SRC has a good performance for recognition under occlusion. In the second experiment on ORL face database, we test the ability of KSRC for handling occlusion. For the last image of every individual, the region from 30 to 60 in width and from 30 to 60 in length was replaced by a $31 \times 31$ black block. Fig. 9 shows the image of one person under occlusion. We use the first nine images per person for training and the last image under occlusion for testing. SRC and KSRC are used for classification after PCA transformation. The maximal recognition rates and the corresponding dimensions of two classifiers are given in Table 3. From Table 3, it can be seen that KSRC outperforms SRC. Especially for

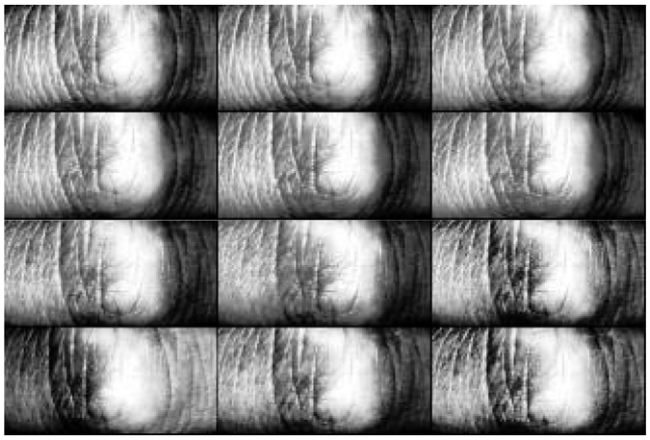

**Fig. 5.** Sample images of one right index finger on PolyU FKP database.

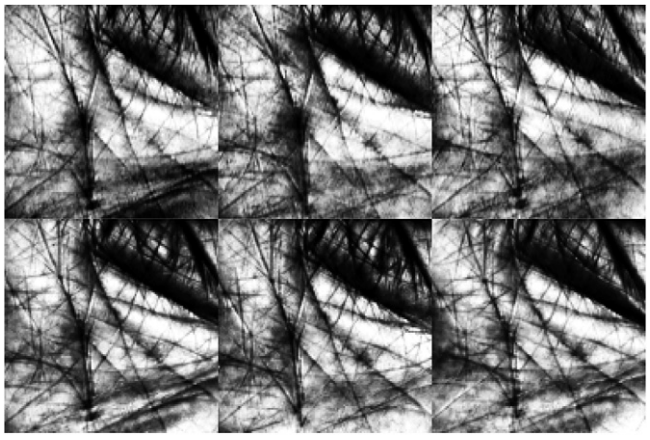

**Fig. 6.** Sample images of one palm on PolyU palmprint database.

**Table 1**
The maximal recognition rates (percent) of NM, NN, KERNEL-NN, SRC and KSRC on FERET face database and the corresponding dimensions and parameters.

| Method | NM | NN | KERNEL-NN | SRC | KSRC (polynomial) | KSRC (Gaussian) |
|---|---|---|---|---|---|---|
| Recognition rate | 40.6 | 54.0 | 54.6 | 55.8 | 58.4 | 56.6 |
| Dimension | 210 | 150 | 250 | 150 | 210 | 210 |
| Parameter | – | – | $d=0.8$ | – | $d=2$ | $t=2$ |

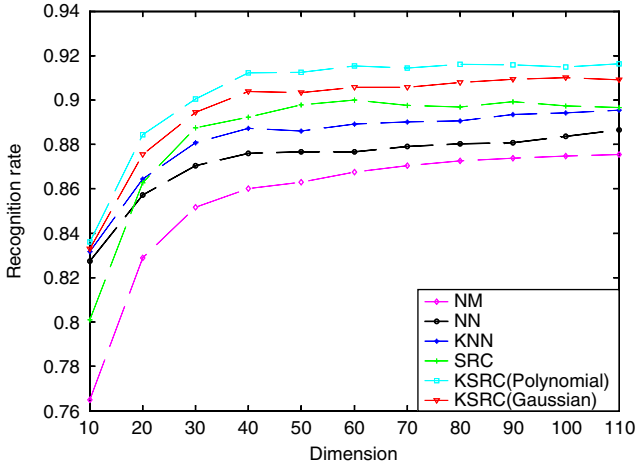


**Fig. 8.** The average recognition rates of NM, NN, KERNEL-NN, SRC and KSRC versus the dimensions on ORL face database across 10 runs.

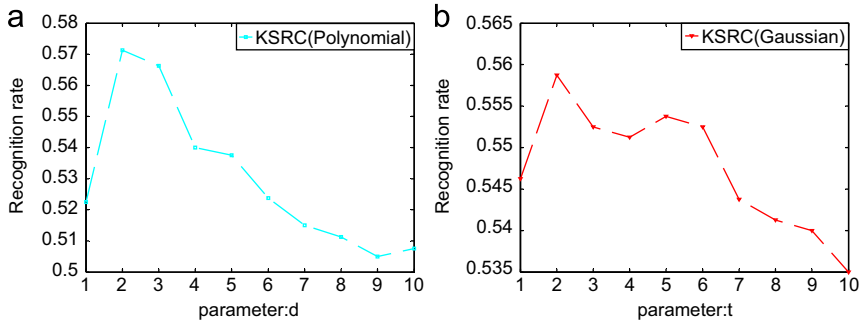


**Fig. 7.** The recognition rates of KSRC on FERET face database versus the variation of the kernel parameters: (a) Recognition rates versus the parameter $d$ of polynomial kernel and (b) recognition rates versus the parameter $t$ of Gaussian kernel.

**Table 2**
The maximal average recognition rates (percent) and standard deviations of NM, NN, KERNEL-NN, SRC and KSRC on ORL face database across 10 runs and the corresponding dimensions and parameters.

| Method | NM | NN | KERNEL-NN | SRC | KSRC (polynomial) | KSRC (Gaussian) |
|---|---|---|---|---|---|---|
| Recognition rate | $87.5 \pm 2.8$ | $88.6 \pm 3.0$ | $89.6 \pm 3.0$ | $90.0 \pm 2.7$ | $91.6 \pm 2.8$ | $91.0 \pm 2.8$ |
| Dimension | 110 | 110 | 110 | 60 | 110 | 100 |
| Parameter | – | – | $d=0.5$ | – | $d=2$ | $t=2$ |

**Fig. 9.** Sample image of one person under occlusion on ORL face database.

**Table 3**
The maximal recognition rates (percent) of SRC and KSRC under occlusion on ORL face database and the corresponding dimensions and parameters.

| Method | SRC | KSRC (polynomial) | KSRC (Gaussian) |
|---|---|---|---|
| Recognition rate | 72.5 | 80.0 | 75.0 |
| Dimension | 330 | 130 | 90 |
| Parameter | – | $d=2$ | $t=3$ |

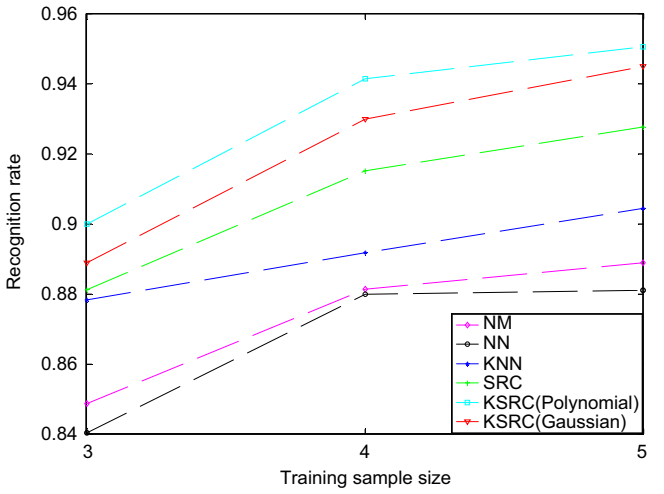


**Fig. 10.** The maximal average recognition rates of NM, NN, KERNEL-NN, SRC and KSRC versus the variation of the training sample size on Yale face database.

KSRC with polynomial kernel, its recognition rate is 7.5 percent more than SRC. Therefore, KSRC is a good classifier to handle occlusion.

On Yale face database, $l$ images per individual ($l$ varies from 3 to 5) are randomly selected for training and the remaining $11-l$ images are used for testing. We run the system 20 times. The first 10 times are used for parameter selection and the rest 10 times are used for performance evaluation of NM, NN, KERNEL-NN, SRC and KSRC. Here RP is used for feature extraction. For RP, samples are projected into a lower dimensional feature space using a random matrix whose column has unit lengths. It has been found to be a sufficiently accurate method for extracting feature of high dimensional data. The optimal kernel parameters are required by global-to-local search strategy. The optimal $d$ of polynomial kernel is set as 5 for KSRC and 0.3 for KERNEL-NN, respectively. The optimal $t$ of Gaussian kernel for KSRC is set as 1. Fig. 10 illustrates the maximal average recognition rates of five methods versus the variation of training sample size. Fig. 10 shows that KSRC still performs best irrespective of the variation of training sample size, whether polynomial kernel is used or Gaussian kernel is used. SRC performs second best. Moreover, KERNEL-NN outperforms NN and NM. These are all consistent with the experiments on FERET and ORL face databases. However, there is also one inconsistent point that NM performs better than NN in this experiment.

For AR face database, the first seven images, which were taken in the first session, are used for training while the rest seven taken in the second session are used for testing. For the PolyU FKP database, we use the first six FKP images collected in the first session for training and the rest six collected in the second session for testing. For the PolyU palmprint database, the first three palmprint images captured in the first session are chosen for training and the remaining three captured in the second session for testing. PCA is first performed for feature extraction and dimension reduction. Then the dimension reduced samples are classified by NM, NN, KERNEL-NN, SRC and KSRC separately. Table 4 lists the maximal recognition rates and the corresponding dimensions of each classification method on three databases. From Table 4, we can see that SRC and KSRC still outperform other three classifiers and KSRC performs better than SRC. This demonstrates that KSRC is more effective than SRC for classification again.

**Table 4**
The maximal recognition rates (percent) of NM, NN, KERNEL-NN, SRC and KSRC on AR face, the PolyU FKP and the PolyU palmprint databases and the corresponding parameters (in parentheses).

| | NM | NN | KERNEL-NN | SRC | KSRC (polynomial) | KSRC (Gaussian) |
|---|---|---|---|---|---|---|
| AR | 53.2 | 66.4 | 66.7 ($d=0.5$) | 73.0 | 74.2 ($d=2$) | 73.7 ($t=5$) |
| PolyU FKP | 42.3 | 58.8 | 59.2 ($d=0.5$) | 66.2 | 70.6 ($d=1$) | 71.8 ($t=9$) |
| PolyU palmprint | 87.3 | 88.3 | 89.0 ($d=0.8$) | 92.7 | 99.3 ($d=1$) | 96.3 ($t=3$) |

## 4. Conclusions

SRC applies sparse representation coefficient to classification. Sparse representation coefficient contains very important discriminating information, so SRC has more powerful discriminating ability than classification methods such as NM, NN and KERNEL-NN. In this paper, we develop a kernel sparse representation based classification (KSRC) algorithm. For KSRC, samples are mapped from original

feature space into a high dimensional feature space first, and then SRC is performed in the high dimensional feature space. Although the explicit samples in the high dimensional feature space are unknown, we prove that SRC could be implemented successfully using kernel function. If an appropriate kernel is utilized, sparse representation coefficient of the test sample in the high dimensional feature space will reflect its label information more accurately. Namely, sparse representation coefficient in the high dimensional feature space contains more effective discriminating information than sparse representation coefficient in the original feature space. Hence, KSRC could obtain higher recognition rate than SRC. Experimental results on FERET, ORL, Yale and AR face databases and the PolyU palmprint and FKP databases indicate the effectiveness of KSRC. For samples containing small noise, SRC and KSRC have the same computational cost. For samples containing big noise, if the number of training sample size is smaller than the dimension of sample, KSRC is more efficient than SRC. Contrarily, SRC is more efficient than KSRC. Besides, when performing KSRC, we should find the optimal parameters. This process will increase its computational cost.

## Acknowledgments

This work is supported by the National Science Foundation of China under Grant nos. 60632050, 60873151, 60973098 and 61005008.

## Appendix A

**The Proof of Theorem 1.** Let

$$\Gamma(\beta) = (B\beta - \phi(y))^T (B\beta - \phi(y))$$

The derivative of $\Gamma(\beta)$ with respect to the variable $\beta$ is

$$\Gamma'(\beta) = 2B^T B\beta - 2B^T \phi(y)$$

Since $\Gamma(\beta) \geq 0$, $\Gamma(\beta)$ achieves the minimum value when

$$B^T B\beta - B^T \phi(y) = 0$$

If $B^T B\beta - B^T \phi(y)$ is close to 0, $\Gamma(\beta)$ will approach its minimum value and

$$\|B\beta - \phi(y)\|_2 = \sqrt{(B\beta - \phi(y))^T (B\beta - \phi(y))} = \sqrt{\Gamma(\beta)}$$

will also approach its minimum value. Therefore, for any $\varepsilon \geq 0$, there must exist $\delta \geq 0$ such that if

$$\|B^T B\beta - B^T \phi(y)\|_2 \leq \delta$$

$\|B\beta - \phi(y)\|_2 \leq \varepsilon$ will be satisfied. □

## References

[1] A.K. Jain, P.W. Duin, J.C. Mao, Statistical pattern recogniton: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
[2] S.B. Kotsiantis, Supervised machine learning: a review of classification techniques, Informatica 31 (2007) 249–268.
[3] K.R. Müller, S. Mike, G. Rätsch, K. Tsuda, B. Schölkopf, An introduction to kernel-based learning algorithms, IEEE Trans. Neural Networks 12 (2) (2001) 181–201.
[4] B. Schölkopf, A. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.
[5] S. Mike, G. Rätsch, J. Weston, B. Schölkopf, K.R. Müller, Fisher discriminant analysis with kernels, in: Proceedings of the 1999 IEEE Signal Processing Society Workshop Neural Networks for Signal Processing, vol. IX, 1999, pp. 41–48.
[6] S. Mike, G. Rätsch, B. Schölkopf, A. Smola, J. Weston, K.R. Müller, Invariant feature extraction and classification in kernel spaces, in: Proceedings of the 13th Annual Neural Information Processing Systems Conference, 1999, pp. 526–532.
[7] K. Yu, L. Ji, X.G. Zhang, Kernel nearest-neighbor algorithm, Neural Process. Lett. 15 (2002) 147–156.
[8] J. Yang, J. Wright, T. Huang, Y. Ma, Image super-resolution as sparse representation of raw patches, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2008.
[9] S. Rao, R. Tron, R. Vidal, Y. Ma, Motion segmentation via robust subspace separation in the presence of outlying, incomplete, and corrupted trajectories, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2008.
[10] J. Mairal, G. Sapiro, M. Elad, Learning multiscale sparse representations for image and video restoration, SIAM MMS 7 (1) (2008) 214–241.
[11] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[12] E. Amaldi, V. Kann, On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems, Theor. Comput. Sci. 209 (1998) 237–260.
[13] E. Candè, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Commun. Pure Appl. Math. 59 (8) (2006) 1207–1223.
[14] D. Donoho, For most large underdetermined systems of linear equations the minimal $l_1$-norm solution is also the sparest solution, Commun. Pure Appl. Math. 59 (6) (2006) 797–829.
[15] E. Candè, T. Tao, Nearest-optimal signal recovery from random projections: universal encoding strategies, IEEE Trans. Inf. Theory 52 (12) (2006) 5406–5425.
[16] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (1) (2001) 129–159.
[17] M. Turk, A.P. Pentland, Face recognition using eigenfaces, in: Proceeding of the IEEE international Conference on Computer Vision and Pattern Recognition, 1991, pp. 586–591.
[18] E. Binham, H. Mannila, Random projection in dimension reduction: application to image and text data, in: Proceeding of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 245–250.
[19] P.J. Phillips, H. Moon, S.A. Rizvi, P.,J. Rauss, The FERET evaluation methodology for face-recognition algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 22 (10) (2000) 1090–1104.
[20] A. Martinez, R. Benavente, The AR face database, CVC Tech. Rep. 24 (1998).
[21] L. Zhang, L. Zhang, D. Zhang, H.L. Zhu, On-line finger-knuckle-print verification for personal authentication, Pattern Recognition 43 (7) (2010) 2560–2571.
[22] D. Zhang, Palmprint Authentication, Kluwer Academic, 2004.
[23] D. Donoho, Y. Tsaig, Fast solution of $l_1$-norm minimization problems when the solution may be sparse, Preprint, 〈http://www.stanford.edu/~tsaig/research.html〉, 2006.

**Jun Yin** received BS degree in Mathematics and PhD degree in Pattern Recognition and Intelligence System from Nanjing University of Science and Technology in 2006 and 2011, respectively. His current research interest includes pattern recognition, face recognition and machine learning.

**Zhonghua Liu** received BS degree in Computer Engineering from the First Aeronautical Institute of the Air Force, MS degree in Computer Software and Theory from Xihua University and PhD degree in Pattern Recognition and Intelligence System from Nanjing University of Science and Technology in 1998, 2005 and 2011, respectively. His current research interest includes pattern recognition, face recognition and image processing.

**Zhong Jin** received BS degree in Mathematics, MS degree in Applied Mathematics and PhD degree in Pattern Recognition and Intelligence System from Nanjing University of Science and Technology in 1982, 1984 and 1999, respectively. Now he is a professor in the Department of Computer Science and Technology at Nanjing University of Science and Technology. His current interest includes pattern recognition, image processing and face recognition.

**Wankou Yang** received BS degree in Computer Science and Technology, MS degree and PhD degree in Pattern Recognition and Intelligence System from Nanjing University of Science and Technology in 2002, 2004 and 2009, respectively. Now he is a Postdoctoral Fellow in the school of automation at Southeast University. His research interest includes pattern recognition, computer vision and digital image processing.