Pattern Recognition 76 (2018) 1–13

Contents lists available at ScienceDirect

Pattern Recognition

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

Virtual dictionary based kernel sparse representation for face recognition

Zizhu Fan[a,b,*], Da Zhang[b], Xin Wang[b], Qi Zhu[c], Yuanfang Wang[b]

[a] School of Basic Science, East China Jiaotong University, Nanchang, China
[b] Department of Computer Science, University of California, Santa Barbara, CA, USA
[c] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

ARTICLE INFO

Article history:
Received 3 June 2016
Revised 30 September 2017
Accepted 3 October 2017
Available online 14 October 2017

Keywords:
Kernel sparse representation for classification (KSRC)
Virtual dictionary
Coordinate descend
Face recognition

ABSTRACT

Kernel sparse representation for classification (KSRC) has attracted much attention in pattern recognition community in recent years. Although it has been widely used in many applications such as face recognition, KSRC still has some open problems needed to be addressed. One is that if the training set is of a small scale, KSRC may potentially suffer from lack of training samples when a nonlinear mapping is used to transform the original input data into a high dimensional feature space, which is often accomplished using a kernel-based method. In order to address this problem, this work proposes a scheme that automatically yields a number of new training samples, termed virtual dictionary, from the original training set. We then use the yielded virtual dictionary and the original training set to build the KSRC model. To improve the computational efficiency of KSRC, we exploit the coordinate descend algorithm to solve the KSRC model. Our approach is referred to as kernel coordinate descent based on virtual dictionary (KCDVD). KCDVD is easy to implement and is computationally efficient. Experiments on many face databases show that the proposed algorithm is effective at remedying the problem with small training samples.

1. Introduction

In recent years, sparse representation for classification (SRC) [1,2] has attracted much attention in machine learning and pattern recognition community, particularly for face recognition [3–5]. It is well known that the typical SRC method contains two main steps. The first step is the sparse representation (SR). Given a sample or data point, SR exploits some or all training samples or data points to represent this sample (data point) based on one or more vector norm minimization, e.g., L_1 norm minimization, which leads to the representation coefficients that are sparse. Here sparse coefficients mean that most representation coefficients are zeros or approaching zeros. The second step is the classification. That is, SRC uses the representation results to classify the test samples. SRC can achieve robust classification effectiveness, since it can deal well with noisy data. In face recognition, SRC has been shown to be a robust method to classify face image data with corruption and occlusion [3].

Besides L_1 norm minimization, there are a number of norm minimization methods to produce sparse coefficients in SRC based approaches. Actually, SRC is essentially based on L_0 norm minimization [4]. L_0 norm means the number of the non-zero entries in a vector. Although it is not really a vector norm in a mathematics sense, the L_0 norm minimization can lead to the sparsest representation. In [5], Xu et al. proposed a SRC method based on $L_{1/2}$ norm minimization which is performed by iterative L_1 norm minimization. In addition, $L_{2,1}$ can also yield the sparse representation [6,7]. In general, directly applying L_2 norm minimization cannot yield the sparse representation [8]. Nevertheless, Xu et al. proposed a two phase test sparse representation (TPTSR) for face recognition [9] using a modified L_2 norm minimization, which was shown to produce a sparse representation and achieve good recognition results.

A typical SRC algorithm is performed in the original input space. It is widely believed that standard SRC algorithms cannot capture the nonlinear information within the data, especially for high-dimensional data sets such as face image databases [10]. In order to capture the nonlinear information within the data, Gao et al. proposed a kernel sparse representation for classification (KSRC) method [11]. They first applied a nonlinear mapping to transform the input data into a high dimensional (even infinite dimensional) feature space that is called the reproducing kernel Hibert space (RKHS) [12]. Then the typical SRC algorithm was per-

* Corresponding author.
E-mail addresses: zzfan@ecjtu.edu.cn, zzfan3@163.com (Z. Fan).

https://doi.org/10.1016/j.patcog.2017.10.001
0031-3203/

formed in this space. KSRC can also theoretically solve the problem that the sample vectors belonging to different classes have very similar, or identical, direction, which the typical SRC algorithm cannot well address without nonlinear warping of data [13].

Kernel sparse representation and its variants have been used in many applications. In [14], Jian proposed the class-discriminative kernel sparse representation-based classification by using multi-objective optimization. Aiming to deal well with the nonlinearity within in multimodal biometrics data, Shekhar developed a kernelized multimodal sparse representation approach [15]. Also, Shrivastava proposed a multiple kernel learning approach to represent the nonlinearity in a high-dimensional RKHS. This approach uses two-step training procedure to learn the kernel weights and sparse representation coefficients [16]. By exploiting the empirical mode decomposition and morphological wavelet-based features, He et al. introduced kernel sparse multitask learning for hyperspectral image classification [17]. In order to perform gesture recognition, Zhou et al. proposed a kernel sparse representation for classifying complicated human gesture. Their approach is robust to the large variability within human gestures [18]. Huang successfully combined sparse representation classifier and kernel discriminant analysis (KDA), and proposed kernel extended dictionary (KED) for face recognition, which was extended to multikernel space to fuse different types of features [19]. Similarly, Zhang proposed a multiple kernel sparse representation approach for face recognition [20], and Gu introduced a multiple kernel sparse representation classification framework for airborne LiDAR data classification [21]. By utilizing the high-dimensional nonlinear information, Feng proposed a kernel combined sparse representation for disease recognition [22]. Based on the assumption that data points belong to Riemannian manifolds, Wu developed a manifold kernel sparse representation of symmetric positive-definite matrices for image classification, face recognition and visual tracking [23]. In [24], Wang applied the kernel sparse representation to the visual tracking task. Our previous work, KTSRC [25], based on L_2 norm minimization, is essentially the kernel version of TPTSR. Hence, KTSRC is a kernel sparse representation method to some extent. Similarly, kernel collaborative representation with locality constrained dictionary (KCRC-LCD) [26] is the kernel version of the collaborative representation based classification (CRC) [8] combining local structure information. Since locality usually leads to sparsity, KCRC-LCD can also be viewed as one type of the kernel sparse representation methods.

As previously mentioned, using the L_0 norm minimization can produce the sparsest representation. However, it is well known that the L_0 norm minimization is an NP hard problem [4]. Nevertheless, if the representation model solution is sufficiently sparse, the L_0 norm minimization can be replaced by the L_1 norm minimization on the condition that each class has sufficient training samples [3]. When the training samples are not enough, the representation model solution may not be sufficiently sparse, and the L_1 norm minimization tends to be unsuitable for the representation model and fails to achieve desirable representation results. Aiming to address this problem, Deng et al. introduced an auxiliary intraclass variant dictionary to the classical SRC model and proposed an extended SRC (ESRC) [27] to deal with the undersampling problem, i.e., each class has a few, or even a single, training samples. Note that the ESRC algorithm is performed in the original input space rather than the kernel induced feature space in which the nonlinear information can be well captured. In addition, the computational efficiency of the ESRC algorithm is very poor, which hinders its application in many real-time applications.

On the other hand, if the training set is of a small scale, mapping the original input data into the high-dimensional feature space may worsen the undersampling problem, since the training set becomes even more sparse in a higher dimensional space. It seems that the kernel version of the above ESRC algorithm may address the undersampling problem in the feature space. However, if the auxiliary intraclass variant dictionary of ESRC in the original input space is mapped into the high dimensional feature space, it is uncertain if the new mapped dictionary can still play the important role to the representation. Besides, the computational complexity may be higher than the original ESRC algorithm. One possible solution is that we can directly generate some new training samples from the original training set to deal with the above undersampling problem. For instance, Zhu proposed a kernel sparse representation approach combing the virtual dictionary obtained from the original training set via Metafaces framework [28]. By adding random noise to original training samples, Tang formed a new training set to perform sparse representation approach [29]. These two methods can achieve good recognition results but do not pay attention on the computational efficiency.

Recently, Xu et al. developed a new algorithm to yield the virtual face images that can effectively improve the typical SRC method for face recognition [30]. Nevertheless, Xu's method fails to capture the nonlinear information within the data, and hence, can be further improved if it considers the nonlinear information. To this end, we will use Xu's method to generate the new training samples and then map them into the RKHS space in our work. The new mapped training samples are referred to as virtual dictionary. Also, the original samples are transformed into the RKHS space. Thus, we can use the virtual dictionary and the mapped training samples to represent and classify the mapped test samples in the high dimensional feature space. Similar to the typical SRC algorithm, the typical KSRC approach is also time consuming. In order to improve the computational efficiency of solving kernel sparse representation (KSR) model in the feature space, we apply the coordinate descent approach [31] to solve the KSR model based on the virtual dictionary. Hereafter, our proposed algorithm is referred to as kernel coordinate descent based on virtual dictionary (KCDVD) approach.

Our KCDVD approach has the following salient properties. First, compared with the typical SRC method, the proposed KCDVD algorithm that exploits the nonlinear mapping can capture the nonlinear information within the data. This property is helpful for correctly classifying the samples. Second, the typical KSRC algorithm suffers from the undersampling problem in the high-dimensional space. Our method yields the virtual dictionary and can alleviate this problem. Third, KCDVD is significantly more efficient than other SRC based methods using auxiliary dictionary derived from the training samples, since our method applies the coordinate descent approach which is a fast scheme to solve the SRC based model. Finally, our method is simple and can be easily implemented. In short, KCDVD is an effective and efficient face recognition algorithm. Extensive experiments on many popular face databases demonstrate that our method is a promising approach.

The remainder of the paper is organized as follows: In Section 2, we describe the related work of KSRC. Section 3 gives the procedure of yielding the virtual dictionary. Section 4 presents the kernel coordinate descent based on the virtual dictionary. Section 5 reports the experimental results and illustrates the effectiveness and efficiency of the proposed algorithms. Section 6 offers our conclusions.

2. Related work

In this section, we briefly review the related works on sparse representation for classification (SRC) and Kernel SRC (KSRC).

2.1. Sparse representation for classification (SRC)

SRC is first proposed to perform face recognition by Wright [3]. Suppose there are N training samples in the original input space $X = [x_1, x_2, ..., x_N] \in R^{d\times N}$and a test sample $y \in R^d$. They are all from c classes. SRC deals with test samples one by one. The typical SRC model is to resolve the following L_1 norm minimization problem.

$$\hat{\alpha} = \min_{\alpha}(\|y - X\alpha\|^2 + \lambda\|\alpha\|_1) \tag{1}$$

where $\alpha = [a_1, a_2, ..., a_N]^T \in R^N$ is the representation coefficient vector and λ is a balance parameter between the sparseness of the coefficient vector and the reconstruction error. If the coefficient vector α is obtained, we compute the representation residual of each class as follows

$$r_k(y) = \|y - X\delta_k(\alpha)\|_2^2,\ (k = 1, 2, ..., c) \tag{2}$$

where c is the number of the classes, and $\delta_k(\alpha) \in R^N$ is a vector whose only nonzero entries are the entries in α associated with class k. Thus, we can classify the test sample y by using the following equation

$$\text{Label}(y) = \arg\min_k r_k(y),\ (k = 1, 2, ..., c) \tag{3}$$

2.2. Kernel SRC (KSRC)

Compared with the typical SRC algorithm, KSRC can capture well the nonlinear information within the data set which is helpful for classification task. It can overcome the drawbacks of SRC [13,32]. The main procedure of KSRC is that all samples are mapped into a high-dimensional RKHS space via a nonlinear mapping. Then, we perform the typical SRC algorithm in this new space and obtain the KSRC learning model. By using a nonlinear mapping, we map the training samples $X = [x_1, x_2, ..., x_N] \in R^{d\times N}$ and the test sample $y \in R^d$ in the original input space into the RKHS space. They are respectively denoted as $\varphi(X) = [\varphi(x_1), \varphi(x_2), ..., \varphi(x_N)] \in R^{D\times N}$ and $\varphi(y) \in R^D$. The kernel sparse representation (KSR) model is as follows

$$\hat{\beta} = \min_{\beta}(\|\varphi(y) - \varphi(X)\beta\|^2 + \mu\|\beta\|_1) \tag{4}$$

Similarly, in the above equation, $\beta = [\beta_1, \beta_2, ..., \beta_N]^T \in R^N$ is the representation coefficient vector and μ is also a balance parameter between the sparseness of the coefficient vector and the reconstruction error. After obtaining the coefficient vector β, we compute the representation residual of each class as follows

$$R_k(y) = \|\varphi(y) - \varphi(X)\delta_k(\beta)\|_2^2,\ (k = 1, 2, ..., c) \tag{5}$$

where $\delta_k(\beta) \in R^N$ is a vector whose only nonzero entries are the entries in β associated with class k. Thus, we can classify the test sample y by using the following equation

$$\text{Label}(y) = \arg\min_k R_k(y),\ (k = 1, 2, ..., c) \tag{6}$$

The above equation is based on the representation residual. Specifically, in the representation of the test sample, if a class achieves the minimal representation residual, the test sample is classified into this class. This scheme is not similar to the support vector machine (SVM) classifier that aims to find the support vectors leading to maximal margin between data classes.

3. Virtual dictionary

In this section, we will introduce the method to generate new training samples- Approximately symmetrical face images (ASFI), proposed by Xu et al. for face recognition [30]. Note that while we use ASFI to generate the virtual dictionary in this article, our recognition scheme is general enough to use other suitable schemes for populating a virtual dictionary.

It is well known that the face of a person is symmetrical or nearly symmetrical. That is, if we know the left half of a face, the right half of the face can be similarly inferred, and vice versa. The ASFI method was proposed based on this fact. Suppose that there is a face image $I \in R^{m\times n}$ where n is even. We first reshape the left half part of this image to a column vector, denoted by I_1. Concretely, the first column of the left half image is changed to the first m entries of the vector I_1, and the second column of the left half image is changed to the second m entries of the vector I_1. This procedure repeats until the last column of the left half image is changed to the last m entries of the vector I_1. Hence, $I_1 \in R^{\frac{mn}{2}\times 1}$. Similarly, the right half of this image is first reversed column-wise and then reshaped into another column vector, denoted by I_2. The aim of the ASFI method is to minimize $\|I_1 - I_2\|^2$. To this end, Xu applied the gradient descent scheme to iteratively minimize it. We denote the initial values of I_1 and I_2, as I_1^0 and I_2^0, respectively. Then, they are iteratively updated as follows

$$I_1^{t+1} = I_1^t - \xi(I_1^t - I_2^t) \tag{7}$$

$$I_2^{t+1} = I_2^t - \xi(I_2^t - I_1^t) \tag{8}$$

where ξ is the learning rate and I_1^t and I_2^t are the values of I_1 and I_2 at time t, respectively. The updating procedure repeats until $\|I_1 - I_2\|^2 \le \delta$, where δ is a small positive constant, or when the number of iterations is larger than a preset value. After obtaining the final vectors I_1^t and I_2^t, ASFI reshapes them to two images and reverses column-wise the image corresponds to the vector I_2^t. Finally, a new virtual face image is produced by juxtaposing these two images. That is, the image associated with I_1^t is the left half of the new image, and the image associated with I_2^t is the right half of the new face image.

Fig. 1 shows some examples of the virtual images obtained by the ASFI method. Fig. 1(a) gives some original face images from the face database ORL that has 40 persons each providing 10 images [33,34]. Fig. 1(b) shows the corresponding virtual images yielded from the original face images by using the ASFI method. From this figure, we can see that although the virtual images are slightly different from their associated original face images, the virtual images and their corresponding original face images appear to be from the same class (person). According to the theory of the SRC method, all images or samples in a class should span a subspace associated with this class. Therefore, the virtual face images and their corresponding images are located in the same subspace associated with their class. Except for large pose variants, the ASFI method is robust to different illuminations, expressions and small pose variants in principle. In other words, the virtual images obtained by the ASFI method are often located in the subspace spanned by the class to which those images, including the generated virtual images and their corresponding face images, belong. From the viewpoint of the sparse representation, the denser subspace of each class tends to lead to the sparser solution to the sparse representation model [3], and hence, we expect to achieve better classification results using the enhanced sparse representation.

Essentially, ASFI can be viewed as one type of data augmentation approaches that construct sampling algorithms by introducing the unobserved data [35]. Unlike the marginal and conditional augmentation methods in [35] that are based on statistical distribution, ASFI is based on the geometric structure of images and is very suitable for dealing with symmetrical data. ASFI may also generate abnormal images from the original face images, particularly when the data are not symmetrical. These abnormal images can be viewed as noise in the SRC or KSRC algorithm. It is well known

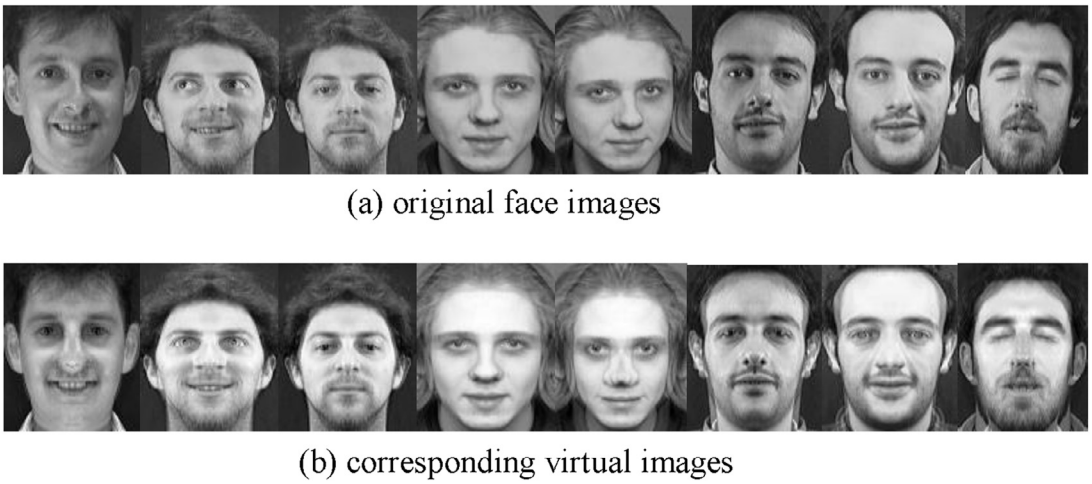


Fig. 1. Original face images and their corresponding virtual images. (a) Original face images; (b) corresponding virtual images.

that SRC and KSRC are robust to the noises since it is based on the L_1 norm minimization [1,36]. Therefore, the abnormal images or noising images can hopefully be tolerated in the SRC and KSRC algorithms, as long as the process does not yield too many such bad images. In our method, we exploit nonlinear mapping to transform the virtual images into the high dimensional feature space. These mapped virtual images are referred to as the virtual dictionary. For a test sample image, our method uses the virtual dictionary combining the mapped training samples to represent this test sample and build a new representation model. This model is presented in the following section.

4. Kernel sparse representation using virtual dictionary

The KSR model in Section 2 may achieve desirable results if the training samples are sufficient. Nevertheless, we deal with the case where training samples are not sufficient in face recognition. Moreover, the nonlinear mapping in KSR usually worsens the situation that training samples are insufficient. Increasing the training samples is a natural way to address the above problem of insufficient training samples. Therefore, besides $\varphi(X)$, we use the virtual dictionary that has M training samples to represent the test sample $\varphi(y)$ in the RKHS space. Thus, we have $M+N$ training samples, and they are denoted by $\varphi(X^*) = [\varphi(x_1^*), \varphi(x_2^*), ..., \varphi(x_{M+N}^*)] \in R^{D\times(M+N)}$. Then, the new KSR model using the virtual dictionary is as follows

$$\hat{\theta} = \min_{\theta}(\|\varphi(y) - \varphi(X^*)\theta\|^2 + \mu\|\theta\|_1) \tag{9}$$

where $\theta = [\theta_1, \theta_2, ..., \theta_{M+N}]^T \in R^{M+N}$ is the representation coefficient vector and μ is also a balance parameter between the sparseness of the coefficient vector and the reconstruction error.

Note that the typical kernel sparse representation based on the L_1 norm minimization is computationally inefficient. In order to improve the computational efficiency of this method, we apply the coordinate descent scheme to infer representing samples in the feature space. In our method, we exploit the kernel trick [37,38] which is a well-known technique widely used in the kernel-based methods. By using this technique, we do not need to explicitly specify the nonlinear mapping and can use a suitable kernel function to express the inner product of two samples in the RKHS space. That is, given two samples $\varphi(x)$ and $\varphi(z)$ in the feature space, their inner production is $\varphi(x)^T\varphi(z) = k(x,z)$ where $k(x,z)$ is a kernel function, e.g., the Gaussian kernel function: $k(x,z) = \exp(-\|x-z\|^2/2\sigma^2)$ where σ is the Gaussian kernel width required to be specified in practice.

The above Eq. (9) can be reformulated as

$$\hat{\theta} = \min_{\theta}\left(\left\|\varphi(y) - \sum_{i=1}^{M+N}\theta_i\varphi(x_i^*)\right\|^2 + \mu\|\theta\|_1\right) \tag{10}$$

Note that in the AFSI method we use, $M = N$. By using the kernel trick, we obtain

$$\hat{\theta} = \min_{\theta}(k(y,y) - 2k(\cdot,y)^T\theta + \theta^T K\theta + \mu\|\theta\|_1) \tag{11}$$

where $k(\cdot,y) = (k(x_1^*,y), k(x_2^*,y), ..., k(x_{2N}^*,y))^T$, and

$$K = \begin{pmatrix} k(x_1,x_1) & \cdots & k(x_1,x_{2N}) \\ \vdots & \ddots & \vdots \\ k(x_{2N},x_1) & \cdots & k(x_{2N},x_{2N}) \end{pmatrix}.$$

Next, we will solve Eq. (11) and obtain the coefficient vector θ. Aiming to improve the computational efficiency, we adopt the coordinate descent [24,31] scheme to solve Eq. (11). We first define the cost function

$$J(\theta) = k(y,y) - 2k(\cdot,y)^T\theta + \theta^T K\theta + \mu\|\theta\|_1 \tag{12}$$

Then, the partial derivative of $J(\theta)$ with respect to θ_i ($i = 1, 2, ..., 2N$) is computed as

$$\frac{\partial J(\theta)}{\partial\theta_i} = 2\sum_{j=1}^{2N}\theta_j k(x_j^*, x_i^*) - 2k(x_i^*, y) + \mu\,\mathrm{sgn}(\theta_i) \tag{13}$$

In this work, we use the Gaussian kernel function. Thus, for any vector $\varphi(x)$ of the feature space, $\varphi(x)^T\varphi(x) = k(x,x) = 1$. We set $\frac{\partial J(\theta)}{\partial\theta_i}$ to 0, and have

$$\theta_i = k(x_i^*, y) - \sum_{j=1, j\neq i}^{2N}\theta_j k(x_j^*, x_i^*) - \frac{\mu}{2}\mathrm{sgn}(\theta_i) \tag{14}$$

Let

$$w_\theta(x_i) = k(x_i^*, y) - \sum_{j=1, j\neq i}^{2N}\theta_j k(x_j^*, x_i^*) \tag{15}$$

Then, we obtain

$$\theta_i = w_\theta(x_i) - \frac{\mu}{2}\mathrm{sgn}(\theta_i) \tag{16}$$

Thus, updating coefficient θ_i is independent of all other coefficients θ_j ($j \neq i$). The coefficient θ_i is computed as

$$\theta_i = \mathrm{sgn}(w_\theta(x_i))\left[|w_\theta(x_i)| - \frac{\mu}{2}\right]_+ \tag{17}$$

Algorithm 1 KCDVD algorithm.

1. **Input**: training set $X = [x_1, x_2, ..., x_N] \in R^{d\times N}$, and testing set $Y = [y_1, y_2, ..., y_T] \in R^{d\times T}$.
2. **Initialize**: Compute the matrix K in Eq. (11), and Q in Eq. (22), and the number of iterations m.
3. for t = 1 to T
   Compute $\theta_{in} = QK(\cdot, y_t)$
   for j = 1 to m
   for i = 1 to $2N$
   Compute $w_\theta(x_i)$ by using Eq. (15) and Update θ_i by using Eq. (16);
   end
   end
4. Calculate the residuals $r_k(y_t)$ $(k = 1, 2, ..., c)$ by using Eq. (20);
5. Assign the class label of y_t by using Eq. (21).
6. end

where the function $[\cdot]_+$ is defined as

$$[q]_+ = \begin{cases} q, & q > 0 \\ 0, & q \le 0 \end{cases} \tag{18}$$

After obtaining the coefficients θ_i $(i = 1, 2, ..., 2N)$, we can exploit them to classify the testing sample y in the original space. Then, we need to compute the following residuals

$$r_k(y) = \|\varphi(y) - \varphi(X^*)\delta_k(\theta)\|_2^2, \ (k = 1, 2, ..., c) \tag{19}$$

where c is the number of the classes, and $\delta_k(\theta) \in R^{2N}$ is a vector whose only nonzero entries are the entries in θ associated with class k. Then, Eq. (19) can be rewritten as

$$r_k(y) = 1 - 2k(\cdot, y)\delta_k(\theta) + \delta_k^T(\theta)K\delta_k(\theta) \tag{20}$$

Thus, we can classify the testing sample y by using the following equation

$$\text{Label}(y) = \arg\min_k r_k(y), \ (k = 1, 2, ..., c) \tag{21}$$

In our approach, the coefficient vector θ is updated iteratively by Eq. (17). The initialization of vectors θ is computed as

$$\theta_{in} = (K + \varepsilon I)^{-1}K(\cdot, y) = QK(\cdot, y) \tag{22}$$

where ε is a small positive value, I is the identity matrix and $Q = (K + \varepsilon I)^{-1}$. Our KCDVD is summarized in Algorithm 1.

In our KCDVD algorithm, the number of iterations m is usually a small integer, say, 5 to 10, that easily leads to a sparse solution. In the following experiments, m is set to 5.

In order to intuitively show the effectiveness of KCDVD, we also take the ORL database as an example. ORL contains 40 persons and each person has 10 face images. Here, ORL is divided into two parts. One part is the training set, and another part is the test set. We choose the first three faces of each person for training, and the remaining faces of each person for testing. Fig. 2 shows two test samples (the fifth and tenth images of the first person, i.e., Class 1) and their corresponding representation residuals of each class, obtained by our KCDVD and the typical SRC algorithms. For the first test image shown in Fig. 2(a), both KCDVD and SRC correctly classify this image into Class 1. Nevertheless, from Fig. 2(c) and (d), we can see that KCDVD can achieve the less representation residual of Class 1 than SRC. The least residual obtained by KCDVD is 0.2539, and the largest one is 1.1489. Note that the least residual obtained by SRC is 0.5430 and the largest one is 1.0887. From the point of view of the representation residual, the classification ability of KCDVD is stronger than that of SRC. From Fig. 2(e) and (f), we observe that KCDVD correctly classifies the second test sample shown in Fig. 2(b), whereas SRC fails to correctly classify this sample (classify it into Class 17). This case can further demonstrate that the classification effectiveness of KCDVD is better than SRC.

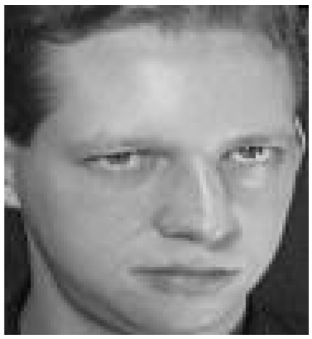
5. Experiments

In order to demonstrate the recognition effectiveness and representation efficiency of our proposed KCDVD algorithm, we conducted many experiments on four popular real-world face databases. The face databases are the GT [30], FERET [39], LFW [40] and CMU PIE [41] databases. For each database, we use a general cross validation scheme [42] for experimentation. That is, we randomly choose some portion of the available samples for training, and the rest of the samples are used for verification. This procedure repeats 10 times for estimating the recognition rates of all experimental methods on each dataset. This technique has the merit that randomly choosing the training set ensures that the classification results will be unbiased [3].

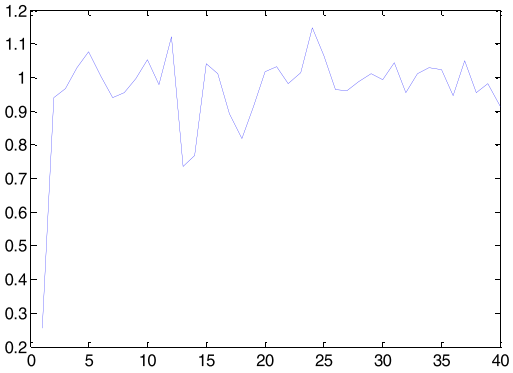
For comparison, we have implemented many classical state-of-the-art representation-based classification schemes: SRC [3], CRC [8], KSRC [13], ESRC [27], kernel coordinate descent for classification (KCD) [24], and SRC based on virtual face images (denoted as SRCVF in our experiments) [24]. Except for CRC, all other methods are based on L_1 norm minimization. The minimization of SRC, KSRC and ESRC is implemented by the popular l1_ls modular [43] which can yield robust recognition results. Besides the above representation-based methods, we have selected the principal component analysis (PCA) combining the nearest neighbor classifier (NNC) as the baseline algorithm for comparison. Here, it is denoted as PCA. Also, we have implemented the linear discriminant analysis (LDA) [44] combining NNC method, denoted as LDA in the experiments. In addition, we have implemented SVM with L_1 norm [45–47], denoted as L1_SVM, and its variant based on virtual face images, i.e., the algorithm L1_SVM using virtual face images obtained by ASFI, which is denoted as SVMVF in the experiment section. These two algorithms are implemented by the popular LIBLINEAR modular [47]. Among all algorithms, there are four algorithms that use the virtual training set. They are SVMVF, ESRC, SRCVF and our method, and the last three methods are based on the SRC algorithm. And other seven algorithms use the original training set.

For the KSRC, KCD and our methods, the kernel function used is the Gaussian kernel function and the kernel parameter is set to be $r \times d$, where d is the average Euclidean distance of the training samples and r, referred to as kernel ratio, needs to be carefully tuned on each data set. Here, we select the best kernel ratio from the set {0.0001, 0.001,0.01, 0.1, 1, 10, 100, 1000} which leads to the highest recognition rates. The selection scheme of penalty parameter in L1_SVM and SVMVF is the same as that of kernel ratio. The penalty parameter in these two algorithms is set to 10. The balance parameter λ in Eq. (3) is set to 1e-3. Also, this parameter in SRC and ESRC is equal to 1e-3. In SRCVF, SVMVF and our KCDVD algorithms, the number of the iterations is 30 in the procedure of generating the virtual face images. Note that the SRC, KSRC, SRCVR and ESRC algorithms are computationally inefficient. The dimensionality of all face images in all compared algorithms is reduced using principal component analysis (PCA) in all experiments to save the computational time. For a fair comparison, the image dimensionality in our proposed algorithm is also reduced by PCA.

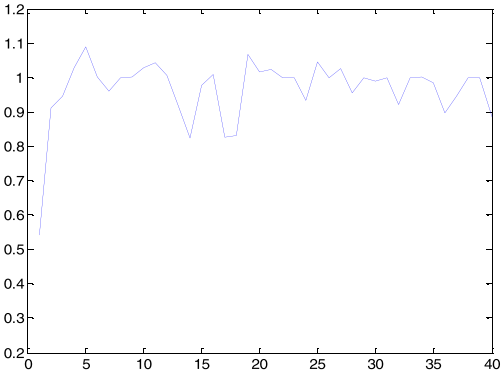
(a) the fifth image of the first person



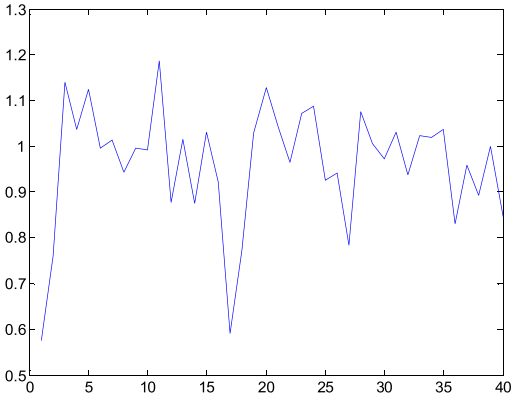
(b) the tenth image of the first person



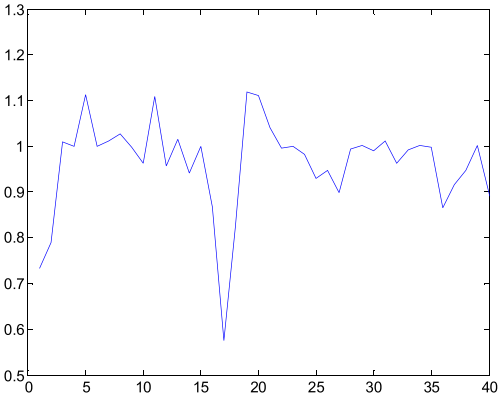
(c) representation residuals of the fifth image by KCDVD



(d) representation residuals of the fifth image by SRC



(e) representation residuals of the tenth image by KCDVD



(f) representation residuals of the tenth image by SRC

Fig. 2. Two samples in ORL and their corresponding representation residuals of each class, obtained by KCDVD and SRC. (a) The fifth image of the first person; (b) the tenth image of the first person; (c) representation residuals of the fifth image by KCDVD; (d) representation residuals of the fifth image by SRC; (e) representation residuals of the tenth image by KCDVD; (f) representation residuals of the tenth image by SRC.

5.1. Experiment on the GT face database

We conducted the first experiment on GT (Georgia Tech) face database. The GT database contains 50 persons with 15 color images per person. The face images of each person characterize several variations such as pose, expression, and illumination [33]. Fig. 3 gives a number of samples of this database. Each image is first changed to grayscale, and then cropped and resized to a resolution of 60×50 pixels. We randomly grouped the image samples of each person into two parts, i.e., the training and testing parts.

Fig. 3. Some face images of GT face database.

Fig. 4. Some image samples of PIE face database.

Table 1
Recognition rates on the GT database.

Algorithms	$N=2$	$N=3$	$N=4$	$N=5$
PCA	43.32±2.36	50.52±2.76	55.31±1.81	59.46±2.12
LDA	33.45±3.56	46.65±2.51	57.07±2.92	64.12±1.78
SRC	46.31±1.31	54.67±2.67	58.53±1.82	62.90±1.93
CRC	46.98±1.39	54.47±2.94	57.38±1.99	61.40±1.89
L1_SVM	43.78±1.13	49.82±1.73	53.45±1.49	58.10±2.27
KSRC	52.18±1.46	59.0±2.18	63.20±2.14	67.78±1.60
KCD	49.26±1.59	58.97±3.65	64.45±1.71	69.22±2.11
SVMVF	46.58±1.55	50.98±2.66	55.55±1.64	59.20±1.85
SRCVF	48.0±1.40	56.12±1.97	59.40±1.84	64.84±1.15
ESRC	48.26±1.12	58.18±3.30	62.91±2.01	67.72±2.58
KCDVD	52.65±1.13	61.98±2.47	68.42±1.42	72.14±2.09

Table 2
Recognition rates on the CMU PIE database.

Algorithms	$N=5$	$N=10$	$N=15$
PCA	36.74±1.69	49.64±1.53	58.52±1.17
LDA	64.41±1.81	80.85±1.86	88.28±0.65
SRC	61.78±1.84	77.13±1.77	84.84±0.92
CRC	62.09±1.76	75.75±1.71	82.65±1.36
L1_SVM	70.18±2.16	82.86±2.05	87.74±0.88
KSRC	66.63±1.61	80.43±1.47	86.33±0.61
KCD	70.49±1.98	82.54±1.65	87.99±1.04
SVMVF	74.70±1.83	86.10±1.46	90.61±0.83
SRCVF	66.53±1.75	79.28±1.31	85.71±0.73
ESRC	68.60±2.08	82.45±1.75	88.46±0.93
KCDVD	74.60±1.62	85.15±1.73	89.72±0.78

For each person, we randomly select a few images (N =2, 3, 4 and 5) for training, and the rest are used for testing.

In this experiment, the dimensionality of the face image data is reduced to 50 using PCA. In the KSRC algorithm, the Gaussian kernel parameter is set to $0.001 \times d$. That is, the kernel ratio is set to 0.001. In the following experiments, i.e., the second, third and fourth experiments, all the kernel ratios in the KSRC method are also set to 0.001. In the KCD and our KCDVD algorithms, the kernel parameter is set to d, i.e., r =1. We ran each algorithm 10 times on each training subset. Table 1 reports the recognition rates (MEAN ± STD-DEV PERCENT) on four training subsets (denoted by $N=2$, 3, 4 and 5 in Table 1).

From Table 1, we can observe that our proposed KCDVD scheme performs the best among the compared methods. Comparing with KCD, KCDVD exploits the virtual dictionary that can yield the sparser representation coefficients, and can achieve more desirable recognition results. Although SRCVF also introduces virtual face images to the typical SRC model, it fails to capture the nonlinear information within the data like KCDVD. As shown in Table 1, KCDVD is better than SRCVF in terms of the recognition rate. Note that the ESRC algorithm also uses the virtual samples to represent the test samples and can address well the undersampling problem in some cases. Nevertheless, since it is directly performed in the original input data space, ESRC cannot capture well the nonlinear information within the data set, which may be the reason why our proposed algorithm is better than the ESRC algorithm.

5.2. Experiment on the CMU PIE face database

The second experiment was conducted on the Carnegie Mellon University (CMU) PIE face database that has 68 individuals. Each individual has photos captured under 13 different poses and 43 different illumination conditions and with four different expressions [41,48]. Fig.4 gives some image samples of CMU PIE face database. This experiment uses the first 30 individuals. All images are manually aligned, cropped and resized to a resolution of 32×32 pixels. We randomly grouped the image samples of each individual into two parts. One part is used for training and the other part is used for testing. The number of training images that is chosen for each individual is 5, 10 and 15, which make up three subsets of the training data.

The face image data dimensionality is reduced to 100 using PCA in this experiment. In the KCD and our KCDVD algorithms, the kernel ratio is set to 3. Also, we run each algorithm 10 times on each training subset and average the results. Table 2 reports the recognition rates on three training subsets. From this table, we can also make a conclusion that our KCDVD algorithm performs significantly better than the SRC algorithm and outperforms other state-of-the-art face recognition methods as a whole (except for SVMVF) in terms of recognition rates.

5.3. Experiment on the LFW face database

We have conducted the third experiment on the labels faces in the wild (LFW) face database that contains more than 13,000 images of faces. LFW investigates the unconstrained face recognition and verification. All the face images of this database are collected from the web [40,49]. We select 946 images of 86 subjects, i.e., 11 images per subject, to perform face recognition experiment. Each image is manually cropped and resized to a resolution of 32×32 pixels. Fig. 5 gives some samples of this database. The number of training images that is chosen for each subject is 7 and 9, which make up two subsets of the training data.

Similarly, the face image data dimensionality is reduced to 200 using PCA in this experiment. The kernel ratio in the KCDVD and KCD algorithms is set to 1 (r =1). Again, we ran each algorithm 10 times on each training subset. Table 3 gives the average recog-

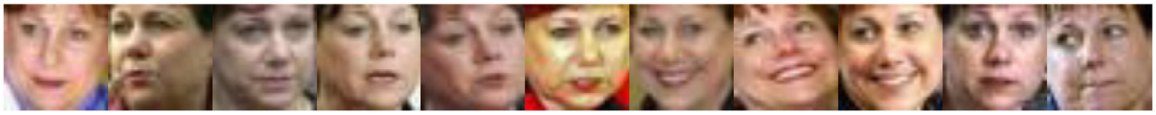


Fig. 5. Some image samples of LFW face database.

Fig. 6. Some image samples of FERET face database.

Table 3
Recognition rates on the LFW database.

Algorithms	$N=7$	$N=9$
PCA	27.91±2.04	30.93±3.56
LDA	24.45±2.24	28.95±2.37
SRC	37.56±1.64	42.33±2.68
CRC	37.15±2.40	42.62±2.81
L1_SVM	34.16±1.87	38.37±2.93
KSRC	40.52±1.76	45.93±2.71
KCD	42.06±2.95	47.50±3.35
SVMVF	34.22±1.21	37.03±3.33
SRCVF	39.80±2.21	45.12±2.78
ESRC	39.74±1.68	46.22±2.92
KCDVD	46.66±2.76	51.63±2.60

Table 4
Recognition rates on the FERET database.

Algorithms	$N=2$	$N=3$
PCA	26.52±1.26	32.38±1.33
LDA	25.57±1.89	47.96±1.27
SRC	29.47±0.69	30.64±1.32
CRC	42.80±1.41	52.84±1.10
L1_SVM	44.02±1.36	55.21±1.13
KSRC	46.66±0.74	58.67±1.29
KCD	48.01±0.99	61.73±1.15
SVMVF	50.10±1.93	62.46±1.31
SRCVF	48.48±1.28	60.50±1.21
ESRC	48.78±1.32	67.17±1.56
KCDVD	53.34±1.11	67.22±1.16

nition rates on the two training subsets. As demonstrated in this table, our KCDVD approach achieves the highest recognition rate, and significantly outperforms other methods.

5.4. Experiment on the FERET face database

The fourth experiment is conducted on the FERET face database in which the images were collected in a semi-controlled environment. We used a subset contains 200 persons and each person has 7 face images. Each of image is resized to a resolution of 40×40 pixels [30,39]. Fig. 6 gives some samples of this database. The number of training images that is chosen for each subject is 2 and 3, which make up two subsets of the training data.

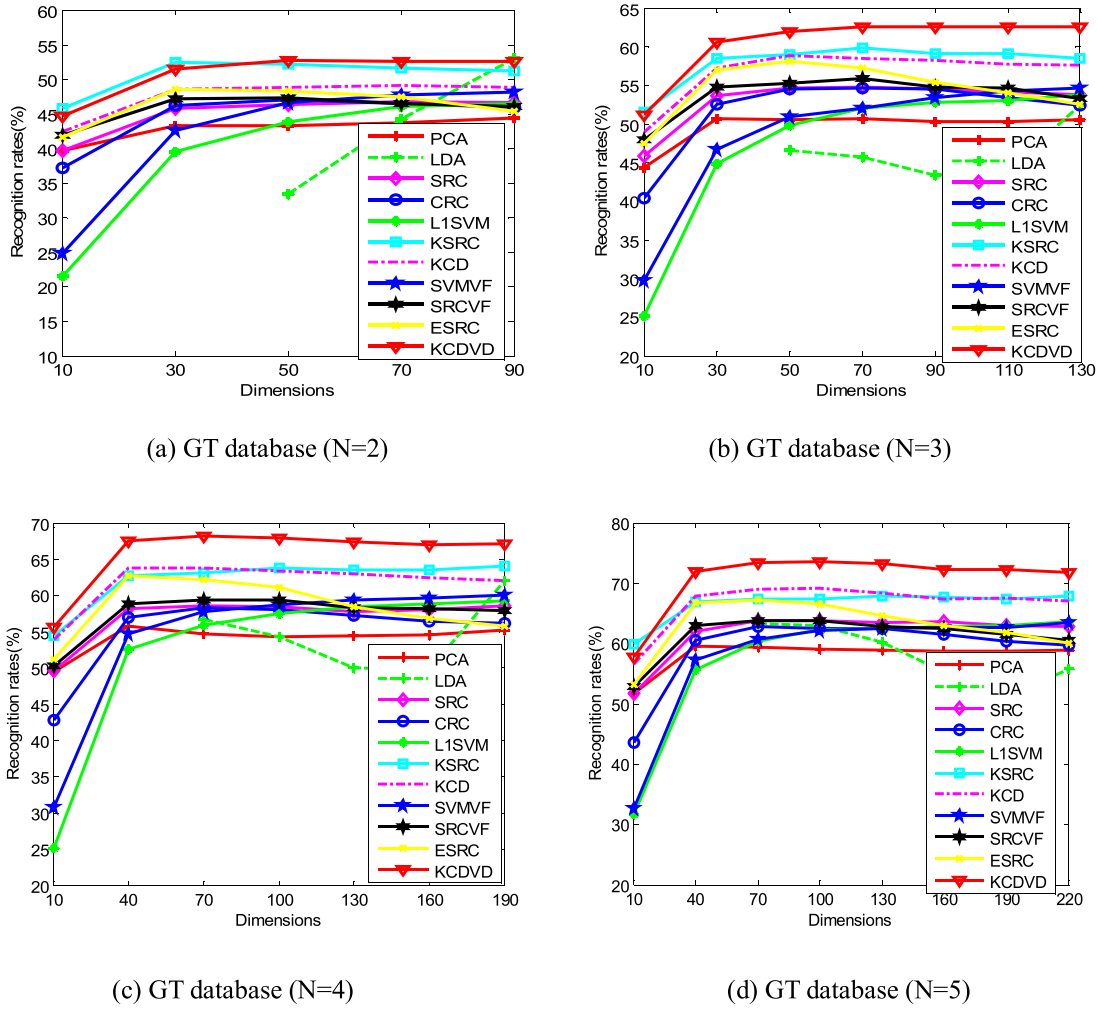
In this experiment, the face image data dimensionality is reduced to 200 using PCA. In the KCD and KCDVD algorithms, the kernel ratio is set to 1. Table 4 reports the average recognition rates on the two training subsets, using 10-fold validation. As expected, we observe from this table that our KCDVD approach is the best method among these state-of-the-art representation-based methods in terms of the recognition rates.

From above experimental results, we can make the conclusion that our KCDVD algorithm achieves superior recognition results. From Tables 1–4, we can also observe that KCDVD tends to perform better when the training set is of small scale in general. This fact verifies that the KCDVD algorithm can usually address the undersampling problem better than other compared algorithms. Moreover, our algorithm is more computationally efficient than the sparse representation based algorithms SRCVF and ESRC that also introduce the virtual samples in the representation models. This is demonstrated in Section 5.6.

5.5. Recognition rates of different dimensions

In order to further investigate the recognition performance of all algorithms, we report their recognition rates of different dimensions in Figs. 7–10, which respectively show the recognition rates on the GT, LFW, FERET and CMU PIE face databases. In each figure, "Dimensions" indicates the dimensions or number of features of each face datum reduced by PCA. Due to the space limitation, we cannot report the recognition rate corresponding to each dimension here. The larger the training set, the more features we used in all algorithms. Since LDA usually requires that the data dimension is not less than the number of data classes, we do not report the recognition rates associated with the data dimensions that are less than the number of data classes on all databases.

From Figs. 7–10, we can observe that our KCDVD method can achieve the best recognition result as a whole. Our method significantly outperforms compared algorithms except for the SVMVF algorithm on the CMU PIE database when $N=5$ and $N=10$. Nevertheless, SVMVF is not stable since it fails to achieve good performance on other face databases. From these figures, we can conclude that SRCVF outperforms SRC, SVMVF outperforms L1_SVM, and our KCDVD outperforms KCD in general. This fact demonstrates that the virtual dictionary used in our method can indeed successfully improve the recognition effectiveness of the traditional algorithms when the training set is small-scale. Note that all algorithms fail to achieve high recognition rates on the LFW and FERET face databases that are well-known complicated face data sets in our experiments. The main reason is that the training set is relatively small and we do not preprocess each face image except for cropping and resizing it. From the point of view of sparse representation, we need sufficient samples to span a subspace if its structure is complicated. In such subspaces, if the scale of the training set is relatively small, it is difficult to effectively represent the test samples by using this training set. Therefore, increasing the number of training samples is an effective method to improve the recognition result on complicated data sets. This is one of motivations of our KCDVD algorithm.



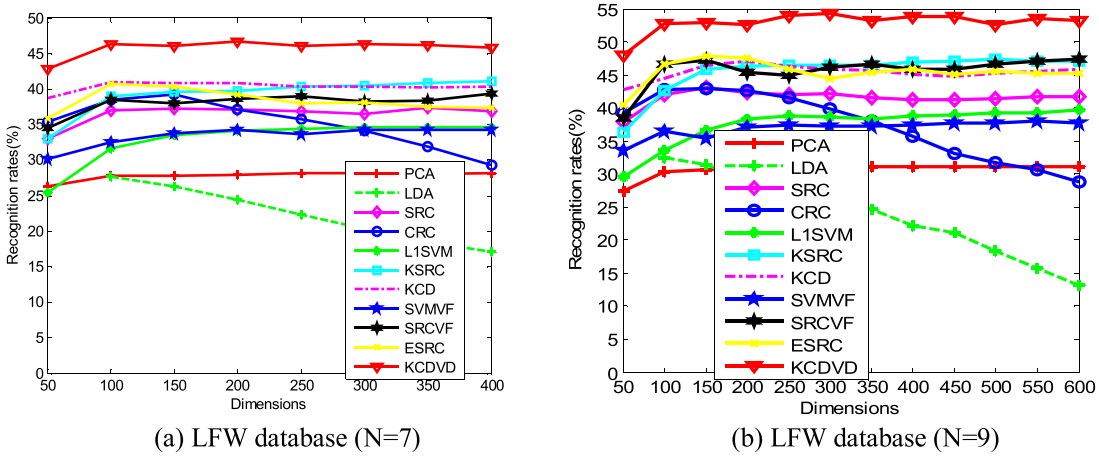
(a) GT database (N=2)

(b) GT database (N=3)

(c) GT database (N=4)

(d) GT database (N=5)

Fig. 7. Recognition rates on the GT database.



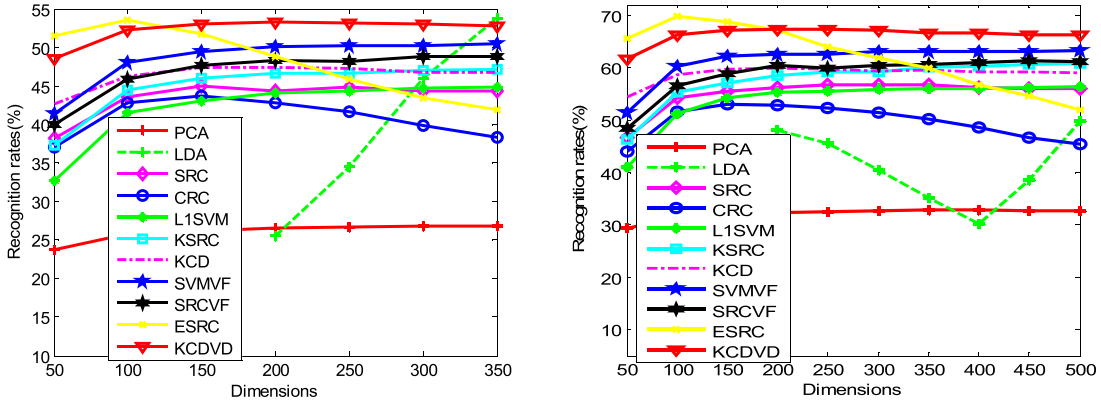
(a) LFW database (N=7)

(b) LFW database (N=9)

Fig. 8. Recognition rates on the LFW database.

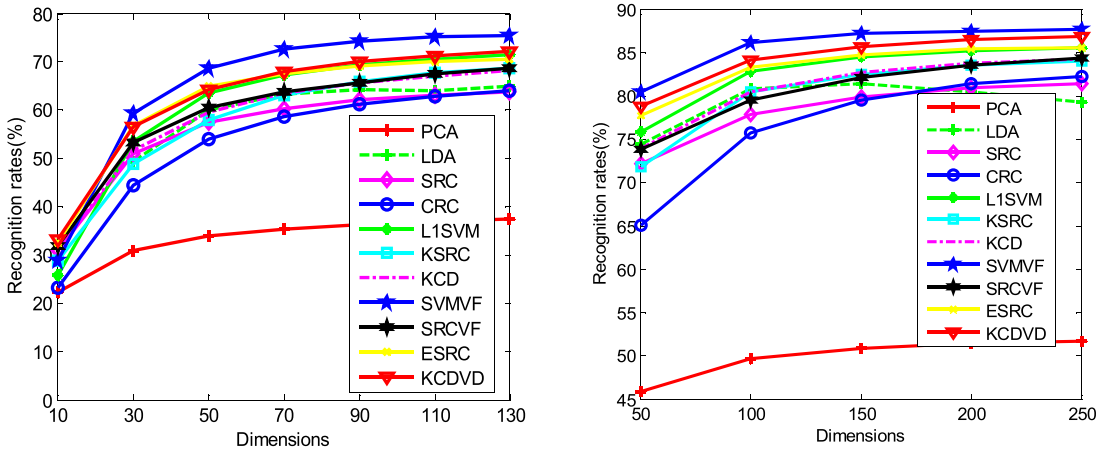
5.6. Computational time of KCDVD

As mentioned above, the SRCVF, ESRC and our KCDVD algorithms are the sparse representation for classification (SRC) based methods, and all of them use virtual samples. Compared with the SRCVF and ESRC algorithms, KCDVD achieves the best recognition results. Moreover, its computational time complexity is the lowest among these three algorithms. To justify this statement, we run these algorithms on each training subset of the databases used in the previous experiments. All the experiments are run on the same platform with Intel Core(TM) i7 2.3 GHz CPU and 8.0GB RAM by Matlab R2011b software. Tables 5–8 report the computational time of the three algorithms on the GT, CMU, LFW and FERET databases, respectively.

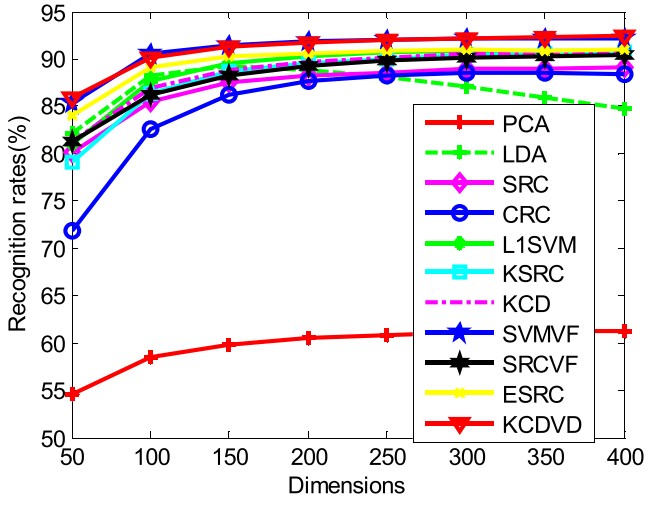


(a) FERET database (N=2) (b) FERET database (N=3)

Fig. 9. Recognition rates on the FERET database.



(a) CMU database (N=5) (b) CMU database (N=10)



(c) CMU database (N=15)

Fig. 10. Recognition rates on the CMU database.

From these tables, we can observe that KCDVD is far faster than the other two algorithms. For example, KCDVD is about 35 times faster than SRCVF when $N=3$ on the GT database. KCDVD needs only about 0.03 seconds to represent and recognize one test sample on this database. In all the experiments, our method needs no more than 1 s to represent and recognize a test sample. As a result, it is easy to adopt our method for real-time applications. By contrast, SRCVF and ESRC spend much more time on representing and recognizing a test sample. For example, ESRC needs about 6 seconds to represent and recognize one test sample on the FERET database when $N=3$.

Table 5
Computational time (s) on the GT database.

Algorithms	$N=2$	$N=3$	$N=4$	$N=5$
SRCVF	559.31	706.87	813.14	841.44
ESRC	502.18	692.28	802.08	827.83
KCDVD	19.01	20.26	34.70	42.45

Table 6
Computational time (s) on the CMU database.

Algorithms	$N=5$	$N=10$	$N=15$
SRCVF	2132.33	3196.43	4087.04
ESRC	2106.53	2221.62	4221.43
KCDVD	217.10	736.46	1622.22

Table 7
Computational time (s) on the LFW database.

Algorithms	$N=7$	$N=9$
SRCVF	457.73	367.16
ESRC	819.57	320.52
KCDVD	182.60	172.69

Table 8
Computational time (s) on the FERET database.

Algorithms	$N=2$	$N=3$
SRCVF	1093.58	1401.19
ESRC	3192.25	4831.32
KCDVD	159.90	314.86

6. Conclusion

In this paper, we proposed the KCDVD algorithm, a new sparse representation based classification approach for face recognition. KCDVD aims to overcome the drawbacks of the classical SRC algorithm. It is performed in the kernel induced feature space and can consequently capture the nonlinear information that is helpful for classification. KCDVD introduces the virtual dictionary to build the sparse representation model more robustly. It can solve the potentially undersampling problem when the training data set is of a small scale and the data dimensionality is very high. As demonstrated by many experiments, our KCDVD method can achieve better recognition rates over the state-of-the-art and has high computational efficiency. The reason for the more accurate recognition results is that KCDVD is performed in the RKHS space by using virtual dictionary. The reason of high computational efficiency is that our method exploits the efficient coordinate descent scheme. Moreover, the implementation of the proposed algorithm is very simple. The basic idea of KCDVD can be conveniently applied to many other applications. In addition, we can consider other approaches to yield more virtual dictionaries. For example, we can exploit small affine transform to distort the images geometrically, or distort the intensity or color values to yield virtual images in the future work.

Acknowledgments

This article is partly supported by Natural Science Foundation of China (NSFC) under grants Nos. 61472138, 61263032 and 61262031, Jiangxi Provincial Natural Science Foundation of China under Grant 20161BAB202066, China's Aviation Science (No. 20145556011), as well as Science and Technology Foundation of Jiangxi Transportation Department of China (2015D0066).

References

[1] J. Yang, L. Zhang, Y. Xu, J.-y. Yang, Beyond sparsity: the role of L1-optimizer in pattern classification, Pattern Recognit. 45 (3) (2012) 1104–1118.
[2] Z. Zhang, Y. Xu, J. Yang, X. Li, et al., A survey of sparse representation: algorithms and applications, IEEE. Transl. Content Mining 3 (2015) 490–530.
[3] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, et al., Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[4] Z. Fan, M. Ni, Q. Zhu, C. Sun, et al., L0-norm sparse representation based on modified genetic algorithm for face recognition, J. Visual Commun. Image Represent. 28 (2015) 15–20.
[5] Z. Xu, X. Chang, F. Xu, H. Zhang, regularization: A thresholding representation theory and a fast solver, IEEE Trans. Neural Networks Learn. Syst. 23 (7) (2012) 1013–1027.
[6] X. Shi, Y. Yang, Z. Guo, Z. Lai, Face recognition by sparse discriminant analysis via joint L 2, 1-norm minimization, Pattern Recognit. 47 (7) (2014) 2447–2453.
[7] C.-X. Ren, D.-Q. Dai, H. Yan, Robust classification using L2, 1-norm based regression model, Pattern Recognit. 45 (2012) 2708–2718.
[8] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition? Presented at the ICCV 2011, Barcelona, Spain, 2011.
[9] Y. Xu, D. Zhang, J. Yang, J.-Y. Yang, A two-phase test sample sparse representation method for use with face recognition, IEEE Trans. Circuits Syst. Video Technol. 21 (9) (2011) 1255–1262.
[10] S. Gao, I.W.-H. Tsang, L.-T. Chia, Kernel sparse representation for image classification and face recognition, in: European Conference on Computer Vision Computer Vision (ECCV 2010), Springer, 2010, pp. 1–14.
[11] S. Gao, I.W.-H. Tsang, L.-T. Chia, Sparse representation with kernels, IEEE Trans. Image Process. 22 (2) (2013) 423–434.
[12] B. Scholkopf, S. Mika, C.J.C. Burges, P. Knirsch, et al., Input space versus feature space in kernel-based methods, IEEE Trans. Neural Networks 10 (5) (1999) 1000–1017.
[13] L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, et al., Kernel sparse representation-based classifier, IEEE Trans. Signal Process. 60 (4) (2012) 1684–1695.
[14] M. Jian, C. Jung, Class-discriminative kernel sparse representation-based classification using multi-objective optimization, IEEE Trans. Signal Process. 61 (18) (2013) 4416–4427.
[15] S. Shekhar, V.M. Patel, N.M. Nasrabadi, R. Chellappa, Joint sparse representation for robust multimodal biometrics recognition, IEEE Trans. Pattern Anal. Mach. Intell. 36 (1) (2014) 113–126.
[16] A. Shrivastava, V. Patel, R. Chellappa, Multiple kernel learning for sparse representation-based classification, IEEE Trans. Image Process. 23 (7) (2014) 3013–3024.
[17] Z. He, Q. Wang, Y. Shen, M. Sun, Kernel sparse multitask learning for hyperspectral image classification with empirical mode decomposition and morphological wavelet-based features, IEEE Trans. Geosci. Remote Sens. 52 (8) (2014) 5150–5163.
[18] Y. Zhou, K. Liu, R.E. Carrillo, K.E. Barner, et al., Kernel-based sparse representation for gesture recognition, Pattern Recognit. 46 (12) (2013) 3208–3222.
[19] K.K. Huang, D.Q. Dai, C.X. Ren, Z.R. Lai, Learning kernel extended dictionary for face recognition, IEEE Trans. Neural Networks Learn. Syst. 28 (5) (2017) 1082–1094.
[20] G. Zhang, H. Sun, G. Xia, Q. Sun, Multiple kernel sparse representation-based orthogonal discriminative projection and its cost-sensitive extension, IEEE Trans. Image Process. 25 (9) (2016) 4271–4285.
[21] Y. Gu, Q. Wang, B. Xie, Multiple kernel sparse representation for airborne LiDAR data classification, IEEE Trans. Geosci. Remote Sens. 55 (2) (2017) 1085–1105.
[22] Q. Feng, Y. Zhou, Kernel combined sparse representation for disease recognition, IEEE Trans. Multimedia 18 (10) (2016) 1956–1968.
[23] Y. Wu, Y. Jia, P. Li, J. Zhang, et al., Manifold kernel sparse representation of symmetric positive-definite matrices and its applications, IEEE Trans. Image Process. 24 (11) (2015) 3729–3741.
[24] L. Wang, H. Yan, K. Lv, C. Pan, Visual Tracking via Kernel Sparse Representation with Multi-kernel Fusion, IEEE Transactions on Circuits and Systems for Video Technology 24 (7) (2014) 1132–1141.
[25] Q. Zhu, Y. Xu, J. Wang, Z. Fan, Kernel based sparse representation for face recognition, in: 21st International Conference on Pattern Recognition (ICPR), 2012, pp. 1703–1706.
[26] W. Liu, Z. Yu, L. Lu, Y. Wen, et al., KCRC-LCD: Discriminative kernel collaborative representation with locality constrained dictionary for visual categorization, Pattern Recognit. 48 (10) (2015) 3076–3092.
[27] W. Deng, J. Hu, J. Guo, Extended SRC: Undersampled face recognition via intraclass variant dictionary, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1864–1870.
[28] J. Zhu, W. Yang, Z. Tang, A dictionary learning based kernel sparse representation method for face recognition, Pattern Recognit. Artif. Intell. 25 (5) (2012) 860–864.
[29] D. Tang, N. Zhu, F. Yu, W. Chen, et al., A novel sparse representation method based on virtual samples for face recognition, Neural Comput. Appl. 24 (3-4) (2014) 513–519.
[30] Y. Xu, Z. Zhang, G. Lu, J. Yang, Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification, Pattern Recognit. 54 (2016) 68–82.

[31] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, J. Statist. Software 33 (1) (2010) 1–22.
[32] J. Yin, Z. Liu, Z. Jin, W. Yang, Kernel sparse representation based classification, Neurocomputing 77 (1) (2012) 120–128.
[33] I. Naseem, R. Togneri, M. Bennamoun, Linear regression for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 2106–2112.
[34] Z. Fan, Y. Xu, D. Zhang, Local linear discriminant analysis framework using sample neighbors, IEEE Trans. Neural Networks 22 (7) (2011) 1119–1132.
[35] D.A. Van Dyk, X.L. Meng, The art of data augmentation, J. Comput. Graphical Statist. 10 (1) (2001) 1–50.
[36] A.Y. Yang, Z. Zhou, A.G. Balasubramanian, S.S. Sastry, et al., Fast-minimization algorithms for robust face recognition, IEEE Trans. Image Process. 22 (8) (2013) 3234–3246.
[37] N. Kwak, Nonlinear projection trick in kernel methods: an alternative to the kernel trick, IEEE Trans. Neural Networks Learn. Syst. 24 (12) (2013) 2113–2119.
[38] A. Shrivastava, V.M. Patel, R. Chellappa, Multiple kernel learning for sparse representation-based classification, IEEE Trans. Image Process. 23 (7) (2014) 3013–3024.
[39] J. Lu, Y.-P. Tan, G. Wang, Discriminative multimanifold analysis for face recognition from a single training sample per person, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 39–51.
[40] Z. Dong, M. Pei, Y. Jia, Orthonormal dictionary learning and its application to face recognition, Image Vis. Comput. 51 (2016) 13–21.
[41] H. Li, L. Zhang, B. Huang, X. Zhou, Sequential three-way decision and granulation for cost-sensitive face recognition, Knowl.-Based Syst. 91 (2016) 241–251.
[42] J.-P. Vert, K. Tsuda, B. Schölkopf, A primer on kernel methods, Kernel Methods Comput. Biol. (2004) 35–70.
[43] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, et al., An interior-point method for large-scale l 1-regularized least squares, IEEE J. Selected Topics Signal Process. 1 (4) (2007) 606–617.
[44] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.
[45] M. Carrasco, J. López, S. Maldonado, A multi-class SVM approach based on the l1-norm minimization of the distances between the reduced convex hulls, Pattern Recognit. 48 (5) (2015) 1598–1607.
[46] Q. Tao, G.-W. Wu, J. Wang, A general soft method for learning SVM classifiers with L1-norm penalty, Pattern Recognit. 41 (3) (2008) 939–948.
[47] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, et al., LIBLINEAR: a library for large linear classification, J. Mach. Learn. Res. 9 (August) (2008) 1871–1874.
[48] J.R. Beveridge, B.A. Draper, J.M. Chang, M. Kirby, et al., Principal angles separate subject illumination spaces in YDB and CMU-PIE, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 351–356.
[49] S.-J. Wang, J. Yang, M.-F. Sun, X.-J. Peng, et al., Sparse tensor discriminant color space for face verification, IEEE Trans. Neural Networks Learn. Syst. 23 (6) (2012) 876–888.

Zizhu Fan received the PhD degree in Computer Science & Technology at Shenzhen Graduate School, Harbin Institute of Technology (HIT), China, in 2014. Now he is an associate professor at School of Basic Science in East China Jiaotong University. Currently, he is visiting the Department of Computer Science, University of California, Santa Barbara, CA, USA. His research interests include pattern recognition and image processing. He has published more than 30 journal papers.

Da Zhang received the BS degree in Information Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014. He is currently working toward the PhD degree in the Computer Vision Laboratory, Department of Computer Science, University of California, Santa Barbara. His research interests include computer vision, pattern recognition and deep learning. He received Outstanding Graduation Award from Shanghai Jiao Tong University in 2014. His personal home page is http://www.cs.ucsb.edu/~dazhang/.

Xin Wang received the BS degree of Computer Science & Technology at Zhejiang University, China, in 2015. He is currently working toward the PhD degree in the Department of Computer Science at University of California, Santa Barbara. His research interests include computer vision, machine learning and deep learning for visual recognition.

Qi Zhu received BS, MS and PhD degrees in Computer Science from Harbin Institute of Technology in 2007, 2010 and 2014, respectively. He is now an assistant professor of College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His interests include pattern recognition and machine learning. He has published more than 20 journal papers.

Yuanfang Wang received MS and PhD degrees in Electrical and Computer Engineering from University of Texas at Austin in 1983 and 1987, respectively. Now he is a full professor in Department of Computer Science, University of California, Santa Barbara. His current interests include pattern recognition and computer vision. From 1998 to 2001, he was an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence. And from 2000 to 2007, he was the associate editor of Pattern Recognition Journal. He was also the program committee of top conferences in pattern recognition and computer vision such as IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) and European Conference on Computer Vision (ECCV).