Kernel Class Specific Centralized Dictionary Learning for Face Recognition

Zhiming Gao, Qian Zhang, Ru Li, Bao-Di Liu and Yanjiang Wang
College of Information and Control Engineering
China University of Petroleum (East China), Qingdao, China
Email: 15763949617@163.com, tyrazhang_426@163.com, 15763944221@163.com,
thu.liubaodi@gmail.com, yjwang@upc.edu.cn

Abstract—**As a fundamental and effective method, sparse representation based classification (SRC) has been applied to computer vision field for many years. However, SRC assumes that the training samples in each class contribute equally to the dictionary which will cause high residual errors and instability. In order to solve the problem and improve classification performance further, class specific centralized dictionary learning (CSCDL) algorithm was proposed. CSCDL considers the concentration of sparse codes in the same class and shows good recognition performance but the fact that CSCDL is only suitable to linear structures limits its applications. To address the limitation, in this paper, we expand the CSCDL algorithm into the kernel space and turn the nonlinear problem into linear ones. Kernel functions and some nonlinear mapping, are used to map original data into a high-dimensional kernel feature space. Effective experimental results on face recognition benchmark databases prove that the performance of kernel class specific centralized dictionary learning (KCSCDL) algorithm is superior to that of CSCDL.**

Keywords*—*kernel space; face recognition; specific centralized dictionary learning

I. Introduction

Nowadays, as one of biometric techniques, face recognition has been applied to many fields by its unique features, such as public security, video surveillance, human-computer interaction, multimedia retrieval and so on. Face recognition usually involves five stages: 1) Face detection, 2) image preprocessing, 3) feature extraction, 4) classifier construction, 5) matching recognition. Many algorithms have been proposed for classifier construction, such as support vector machine (SVM) [1], boosting [2], the nearest neighbor (NN) [3] and so on. Nearest subspace methods were proposed to assign the label of a test image by comparing its reconstruction error for each category [4], [5].

Based on the nearest subspace framework, sparse representation based classification (SRC) was proposed and showed impressive classification performance [6]. SRC represents a test sample with a linear combination of training samples. In each class, some images are selected as training samples, and the rest of images are used as test images. Firstly, sparse coding is carried out on the training sample image set, and then the classification is determined according to the residual error from each class. The experiment proves that the method has strong robustness.

As one of algorithms for sparse representation, dictionary learning [7], [8], [9] has been applied to visual computation areas for many years. Based on the theory of wavelet analysis, Mallat and Zhang put forward the method for adaptive decomposition of signal decomposition [10]. The basic thought is using over-complete dictionary to replace traditional orthogonal basis. In over-complete dictionary, signals can select bases which are used to denote the signal flexibly according to their own characteristics in order to get a certain extent close to the essential characteristics of signals. A large number of experimental data show that the signal sparse representation is more efficient in the over-complete dictionary.

Two models of dictionaries which are used most commonly are comprehensive dictionary model and analytic dictionary model. The former attempts to find a group of base vectors to reflect the signal eigenspace, commonly used methods are the following: Sparsenet dictionary learning algorithm [11] which uses maximum likelihood estimation method to learn dictionary. A lot of small image blocks are extracted from the natural image database as the training set, but its disadvantage is easily being trapped into local optimal. MOD (method of optimal directions) dictionary learning algorithm [12] which uses the OMP algorithm for sparse coding and introduces closed-form solution to update the dictionary. It is assumed that the values of each iteration are not updated one by one but updated together after all iterations. The speed of updating the dictionary is very fast, but it cannot have the global optimal solution due to the limit of the zero norm. K-singular value decomposition (K-SVD) algorithm [13] that is proposed to solve the MOD algorithm's computational complexity. It updates the dictionary one by one, but it also faces the problem to solving sparse coding without global optimal solution. In addition, the algorithm is not guaranteed to converge. Online dictionary learning (ODL) [14] which deals with only one sample each iteration and uses Lars (Least-angle regression) to solve the sparse coding algorithm and the coordinate decline method to update dictionary respectively. This method Can update online, but the processing speed is so slow for large-scale data. The latter considers sparse representation problem from the viewpoint of dual analysis. It tries to find the basis of the orthogonal space of the signal. Compared with the comprehensive model, in the same dimension, analytic model owns more subspaces. The commonly used methods are Analyisis K-SVD [15] algorithm, LST(learning sparsifying transform) algorithm and so on.

978-1-5090-1345-6/16/$31.00 ©2016 IEEE

As one technique to deal with nonlinear data, kernel method was proposed by Vapni and then applied into PCA (principal component analysis) by scho1kopf *et al.* [16]. Although SRC has good robustness, it cannot classify a test sample if it has the same vector direction when training samples belong to two or more classes [17]. In other words, SRC could not deal with the image which includes of nonlinear structures. In order to solve these problems, Zhang *et al.* [17] proposed a kernel sparse representation-based classifier (KSRC) which combines SRC and kernel tricks successfully. In KSRC, the nonlinear data of original space were mapped into a high or even infinite dimensional kernel feature space according to some kernel mapping. As mentioned above, dictionary learning is an efficient method. Zhu, Yang and Tang proposed a dictionary learning based kernel sparse representation method and achieved impressive performance in face recognition [18].

Liu *et al.* [19], [20] proposed class specific dictionary learning (CSDL) approach and class specific centralized dictionary learning (CSCDL) shows the superior performance to SRC. Motivated by the superior performance of CSCDL and the useful kernel tricks, in this paper, we propose a kernel class specific centralized dictionary learning (KCSCDL) algorithm for sparse representation based classification. The main contributions focus on threefold,

1. We extended the CSCDL algorithm to the kernel space to utilize the nonlinear feature property to improve the recognition performance.

2. Hellinger kernel is capable of achieving better performance for face recognition tasks.

3. Experimental results on three databases demonstrate the KCSCDL algorithm is more effective than CSCDL algorithm.

The rest of the paper is organized as follows. Section II reviews class specific centralized dictionary learning algorithm and some related knowledge about kernel tricks. Section III explains our kernel class specific centralized dictionary learning approach for sparse representation. The application of the proposed KCSCDL algorithm in the face recognition is shown and the experimental setup and results analysis are given in Section IV. Finally, discussions and conclusions are drawn in Section V.

II. RELATED WORK

In this section, we review the CSDL algorithm which was proposed in paper [20], [21] and then introduce some knowledge about kernel technique briefly.

A. Overview of CSDL

In SRC, one test sample can be represented as a linear combination of all classes of training samples:

$$y = Bs \tag{1}$$

here, y represents one test sample. B is the dictionary includes of all training samples. $B = [B_1, B_2, ..., B_c]$, where c represents the number of classes. s is a sparse coefficient vector and $s = [0, ..., 0, s_{c,1}, ..., s_{c,n_c}, 0, ..., 0]^T$. N_c is the number of the training samples in the c_{th} class. SRC uses training samples as the dictionary directly, so the test sample can be expressed as the following

$$y = XWs \tag{2}$$

here, the training samples $X = [X_1, X_2, ..., X_c]$, and $W \in R^{N \times N}$ is an identity matrix. This means that the training samples contribute equally for constructing the dictionary $B = XW$ when representing the test sample y. The objective function of CSDL is:

$$\begin{gathered} f(W^c, S^c) = \{||X^c - X^cW^cS^c||_F^2 + 2\alpha \sum_{n=1}^{N^c} ||S_{.n}^c||_1\} \\ s.t. ||X^cW_{.k}^c||_2^2 \leq 1, \forall k = 1, 2, ..., K \end{gathered} \tag{3}$$

here, α is the regularization parameter to control the trade-off between fitting goodness and sparseness, W^c is the learned weight coefficient for constructing the dictionary and S^c is the corresponding sparse representation. c is the c_{th} class and K is the size of the learned dictionary.

CSDL classifies images according to the minimum residual error $id(y)$:

$$id(y) = \arg\min \{||y - XWs||_2^2\} \tag{4}$$

therefore, the final goal is to get the $id(y)$ which will be implemented in the next section.

B. Introduction of kernel method

As an effective technique in machine learning, kernel method's biggest contribution is to map raw data to a high dimensional space so that the nonlinear features of the original space can be dealt with in linear methods. Most of existing image classifiers are linear classifiers which can only deal with linear data, so its application has great limitations in particular. In order to keep the good properties of linear classifiers, the kernel method has become a hot research topic. The following is a brief introduction of the kernel mapping and the kernel function.

1. Kernel mapping: As for the data with nonlinear structure, it is necessary to make data change from a low dimensional space to a high dimensional space with some nonlinear mapping, making the nonlinear structure turn into the linear. Generally, assuming that one original data set $\{x, x \in X, X \in R\}$: and the mapping is concluded as the following:

$$\Phi : X \rightarrow F, x \in X \rightarrow \Phi(X) \in F \tag{5}$$

F is the kernel space.

2. Kernel function: Although we know that the kernel mapping can solve the nonlinear problem, finding a suitable mapping is difficult. KPCA [16] proved that some required projection matrix arithmetic expressions in the original linear space can be represented by the inner product of data in the kernel space through some transformation. Fortunately, the inner product has connection with some function in the original and it is called kernel function κ.

$$\begin{aligned} &\kappa : R^d \times R^d \rightarrow R \\ &\kappa(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >= \Phi(x_i)^T\Phi(x_j) \end{aligned} \tag{6}$$

Therefore, we only need to find the kernel function κ, ignoring the complex operation in kernel space. All kernel functions must satisfy the Mercer conditions.

There are many kernel functions and in this paper, we used five different κ. They are linear kernel, Polynomial kernel, Gaussian kernel, Hellinger kernel and Histogram intersection kernel.

III. KERNEL CLASS SPECIFIC CENTRALIZED DICTIONARY LEARNING

For different classification tasks and various sample distributions, a better dictionary should be more adaptable. Training samples of the same class are ought to have different contributions weight in the corresponding dictionary and other samples from the rest of classes should not work. Based on this, CSDL uses a block-diagonal matrix to replace the W from Eqn. (2). However, for CSDL, sparse codes in different classes are interdependent. Such interdependence among classes may lead to erroneous discrimination. To solve the problem, the centralize sparse codes of the same class and plenty of experimental data showed that the practice has greatly improved the performance.

$$\Re(S^c) = \sum_{n=1}^{N^c} ||S^c_{.n} - E(S^c)||_2^2 \tag{7}$$

here, $E(S)$ represents the mean of each row of a matrix S.

The objective function of CSCDL algorithm is the combine of Eqn. (3) and Eqn. (7).

$$\begin{aligned} f(W^c, S^c) = \{ & ||X^c - X^c W^c S^c||_F^2 + 2\alpha \sum_{n=1}^{N^c} ||S^c_{.n}||_1 \\ & + \eta \sum_{n=1}^{N^c} ||S^c_{.n} - E(S^c)||_2^2 \} \\ & s.t. ||X^c W^c_{.k}||_2^2 \leq 1, \forall k = 1, 2, ..., K \end{aligned} \tag{8}$$

We extend the CSCDL to the kernel space to capture nonlinear structure of the data to improve the classification performance. Thus, the objective function of KCSCDL can be written as follows,

$$\begin{aligned} f(W^c, S^c) = \{ & ||\Phi(X^c) - \Phi(X^c) W^c S^c||_F^2 + 2\alpha \sum_{n=1}^{N^c} ||S^c_{.n}||_1 \\ & + \eta \sum_{n=1}^{N^c} ||S^c_{.n} - E(S^c)||_2^2 \} \\ & s.t. ||\Phi(X^c) W^c_{.k}||_2^2 \leq 1, \forall k = 1, 2, ..., K \end{aligned} \tag{9}$$

Here, η is used for adjusting the trade-off between the reconstruction error and the degree of deviation from the sparse codes to their centers. W^c and W^c are the weight matrix and the sparse codes, respectively. The objective function of Eqn. (9) is not convex to both S^c and W^c. However, if one variable is fixed, Eqn. (9) will be convex to another variable.

Specifically, similar to the optimization strategies adopted in [22], [23], the problem is decomposed into two subproblems via alternating minimization. They are a ℓ_1-norm regularized least-squares minimization problem with fixed W^c and a ℓ_2-norm constrained least-squares minimization problem with fixed S^c.

A. ℓ_1-norm regularized least-squares minimization problem

With fixed W^c , Eqn. (9) will become as the following:

$$\begin{aligned} f(S^c) = & ||\Phi(X^c) - \Phi(X^c) W^c S^c||_F^2 + 2\alpha \sum_{n=1}^{N^c} ||S^c_{.n}||_1 \\ & + \eta \sum_{n=1}^{N^c} ||S^c_{.n} - E(S^c)||_2^2 \end{aligned} \tag{10}$$

Based on the transformation relationship of trace function and F-norm, Eqn. (10) can be simplified as:

$$\begin{aligned} f(S^c) = & -2 \sum_{n=1}^{N^c} [\kappa(X^c, X^c) W^c]_{n.} S^c_{.n} \\ & + \sum_{n=1}^{N^c} S^{cT}_{.n} [W^{cT} \kappa(X^c, X^c) W^c] S^c_{.n} \\ & + 2\alpha \sum_{k=1}^{K} \sum_{n=1}^{N^c} |S^c_{kn}| + \eta \sum_{n=1}^{N^c} (\frac{N^c - 1}{N^c})^2 S^{cT}_{.n} S^c_{.n} \\ & - 2 \frac{N^c - 1}{N^{c2}} S^{cT}_{.n} \sum_{m=1, m \neq n}^{N^c} S^c_{.m} \end{aligned} \tag{11}$$

Optimize the element S^c_{kn} with other elements fixed,

$$\begin{aligned} f(S^c_{kn}) = & S^c_{kn} \{ [W^{cT} \kappa(X^c, X^c) W^c]_{kk} + \eta (\frac{N^c - 1}{N^c})^2 \} \\ & + 2 S^c_{kn} \{ \sum_{l=1, l \neq k}^{K} [W^{cT} \kappa(X^c, X^c) W^c]_{kl} S^c_{ln} \} \\ & - 2\eta S^c_{kn} (\frac{N^c - 1}{N^{c2}} \sum_{m=1, m \neq n}^{N} S^c_{km}) \\ & - 2\eta S^c_{kn} (\frac{N^c - 1}{N^{c2}} \sum_{m=1, m \neq n}^{N} S^c_{km}) \\ & - 2 S^c_{kn} \{ [W^{cT} \kappa(X^c, X^c)]_{kn} \} + 2\alpha |S^c_{kn}| \end{aligned} \tag{12}$$

Here, $[W^{cT} \kappa(X^c, X^c) W^c]_{kk} = 1$, because the restriction of the dictionary which will be explained in ℓ_2-norm. Eqn. (12) can be solved with the convexity and monotonic property of the parabolic function and reaches the minimum at the unique point.

$$\begin{aligned} S^c_{kn} = & \frac{1}{1 + \eta (\frac{N^c - 1}{N^c})^2} \max \{ B_{kn} - [EP^{c^{kn}}]_{kn}, \alpha \} \\ & + \frac{1}{1 + \eta (\frac{N^c - 1}{N^c})^2} \min \{ B_{kn} - [EP^{c^{kn}}]_{kn}, -\alpha \} \end{aligned} \tag{13}$$

Here, $B_{kn} = [W^{cT}\kappa(X^c, X^c)]_{kn} + \eta[\frac{N^c-1}{N^c} \sum\limits_{m=1,m\neq n}^{N^c} S^c_{km}]$, $E = W^{cT}\kappa(X^c, X^c)W^c$, and $P^{c^{kn}} = \left\{ \begin{array}{c} S^c_{pq}, p \neq k||q \neq n \\ 0, p = k\&q = n \end{array} \right.$.

B. ℓ_2-norm constrained least-squares minimization problem

With S^c fixed, Eqn. (9) will become as the following,

$$\begin{aligned} f(W^c) = ||\Phi(X^c) - \Phi(X^c)W^cS^c||_F^2 \\ s.t.||X^cW^c_{.k}||_2^2 \leq 1, \forall k = 1, 2, ..., K \end{aligned} \tag{14}$$

Ignoring the constant term and expressed with the Lagrange function, Eqn. (16) becomes,

$$\begin{aligned} \Gamma(W^c, \mu_k) = &-2\sum_{k=1}^{K}[S^c\kappa(X^c, X^c)]_{k.}W^c_{.k} \\ &+ \sum_{k=1}^{K} W^{cT}_{.k}[\kappa(X^c, X^c)W^cS^cS^{cT}]_{.k} \\ &+ \mu_k(1 - [W^{cT}\kappa(X^c, X^c)W^c]_{kk}) \end{aligned} \tag{15}$$

Here, μ_k is Lagrange multiplier. According to the KKT conditions, the optimum solution must satisfy the two conditions as the following,

$$\begin{aligned} &\frac{\partial\Gamma(W^c, \mu_k)}{\partial W^c_{.k}} = 0 \\ &1 - [W^{cT}\kappa(X^c, X^c)W^c]_{kk} = 0 \\ &\mu_k \neq 0 \end{aligned} \tag{16}$$

Based on KKT conditions, we can get the result,

$$W^c_{.k} = \frac{S^{cT}_{.k} - [Q^{c^k}F]_{.k}}{\sqrt{\left(S^{cT}_{.k} - [Q^{c^k}F]_{.k}\right)^T\kappa(X^c, X^c)(S^{cT}_{.k} - [Q^{c^k}F]_{.k})}} \tag{17}$$

where $F = S^cS^{cT}$ and $Q^{c^k} = \left\{ \begin{array}{c} W^c_{.p}, p \neq k \\ 0, p = k \end{array} \right.$

IV. Experimental results

In this section, we will give the experiment setup and the results analysis. In order to test the recognition performance of KCSCDL algorithm, we carry out the experiment on three benchmark databases, including of the Extended YaleB [24], the AR [25] and the CMU PIE [26].

For every database, each image is cropped into 32×32, forming a column vector with 4096 dimension. In order to eliminate randomness, the experiment is carried out ten times. In each class, 5 images and 10 images are chosen randomly as the training samples and the testing samples, respectively. The column vector is ℓ_2 normalized to form the raw feature.

In this section, to find the optimal kernel, KCSCDL algorithm is tested in 5 kernels on three databases. Parameters of the objective function, like α and η , have great effects on the recognition performance. Therefore, finding best parameters is an important task.

TABLE I
RECOGNITION RATE(%) OF FIVE KERNELS ON THREE DATABASES

Database	linear	Poly	Gaussian	Hellinger	HIK
Extended YaleB	80.60	79.75	74.50	90.90	59.29
CMU PIE	79.16	74.79	65.75	79.13	60.42
AR	92.40	90.72	81.63	85.90	81.10

A. Introducing of databases

1. Extended Yale B database: The extended Yale B database contains 2414 frontal face images of 38 different individuals totally. These images have different illumination conditions, postures and facial expressions, so they are close to the actual application. The experiment is carried out on the whole database with a total of 2414 pictures. Each image has 1024 features.

2. CMU PIE database: The CMU PIE database contains 41368 images of 68 persons with multiple poses, expressions and various illumination conditions. For each class, we choose about 170 images to do the experiment, 11554 images in total.

3. AR database: The AR database is recognized widely. It consists of over 4000 frontal images from 126 individuals. In this experiment, images of 50 males and 50 females are chosen to be tested. Each class has 26 images and 2600 in total.

B. Searching for optimal kernel

Five kernels are tested with the appropriate parameters(α is 2^{-9} and η is 2^{-7}) when the recognition rate is high and stable on three databases. The results are showed in Table I. From Table I, we can see that each database has its optimal kernel. For the Extended YaleB database, the KCSCDL algorithm has the best recognition rate with the Hellinger kernel. On the CMU PIE database, the recognition rate of linear kernel and Hellinger kernel is so close that we cannot judge the best kernel since that it is only one condition which we set. Therefore, in order to find the optimal kernel, both of the two kernels are used to find the optimal parameters. The highest recognition rate on the AR database is the linear kernel.

As mentioned above, we set suitable parameters to find the optimal kernel first. However, the final goal of the experiment is to find the best circumstance, where the KCSCDL algorithm can have the highest recognition rate. Parameters have great effects on algorithm performance. In other word, we should find the best combination of the kernel function and parameters.

C. Matching optimal parameters

There are two parameters α and η which affect each other, so we test one with another one fixed. At first, η is set as 0 and α varies in the range: $\{2^{-1}, 2^{-3}, 2^{-5}, ..., 2^{-11}\}$. After finding the best α, η varies in a suitable range to find the optimal one.

1) Extended YaleB database: For Extended YaleB database, from Table I, the best kernel is Hellinger kernel, so the experiment is based on it. Fig.1 shows the optimal parameters' selection.

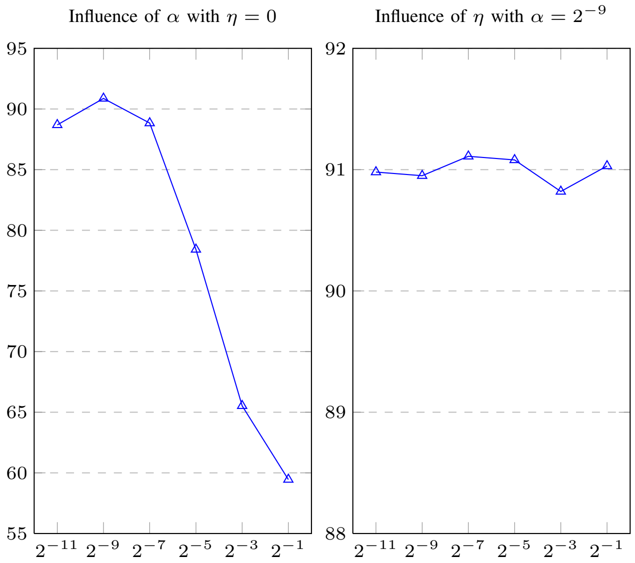

Fig. 1. Influence of α and η on Extended YaleB database for KCSCDL.

From Fig.1, the best α is 2^{-9}. Fixed $\alpha = 2^{-9}$, η is varied in the rage: $\{2^{-1}, 2^{-3}, 2^{-5}, ..., 2^{-11}\}$. When η equals 2^{-7}, the face recognition rate reaches the highest point: 91.11%.

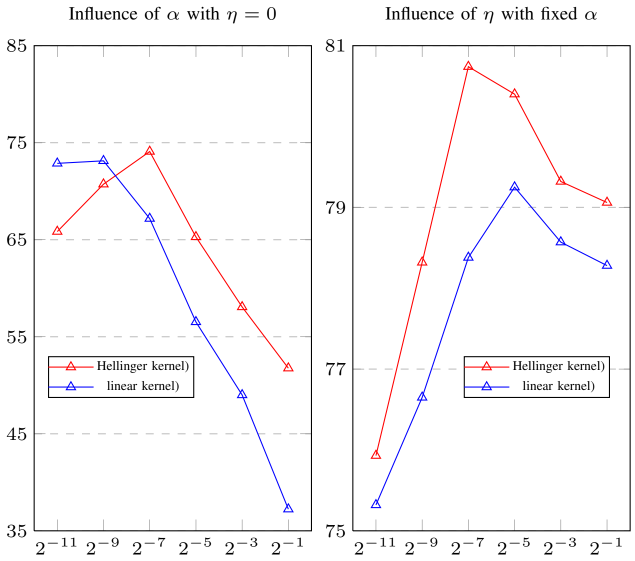

Fig. 2. Influence of α and η on CMU PIE database for KCSCDL. For the right figure, $\alpha = 2^{-9}$ for linear kernel and $\alpha = 2^{-7}$ for Hellinger kernel.

2) CMU PIE database: As discussed above, both of linear kernel and Hellinger kernel should be tested in this part. The results are in Fig.2.

Fig.2 compares the recognition rate in linear kernel and Hellinger kernel when α is varied in range: $\{2^{-1}, 2^{-3}, 2^{-5}, ..., 2^{-11}\}$ with $\eta = 0$. The best α is 2^{-7} for Hellinger kernel and 2^{-9} for linear kernel. Fig.2. shows that the best η is 2^{-7} for Hellinger kernel and 2^{-5} for linear kernel. The highest rate on the CMU PIE database is fetched in the Hellinger kernel, reaching at 80.74%. Therefore, the best combination is the Hellinger kernel with $\alpha = 2^{-7}$ and $\eta = 2^{-7}$.

TABLE II
RECOGNITION RATE(%) OF TWO ALGORITHMS ON THREE DATABASES

Database	CSCDL	KCSCDL
Extended YaleB	80.37	91.11
CMU PIE	78.02	80.74
AR	94.63	94.78

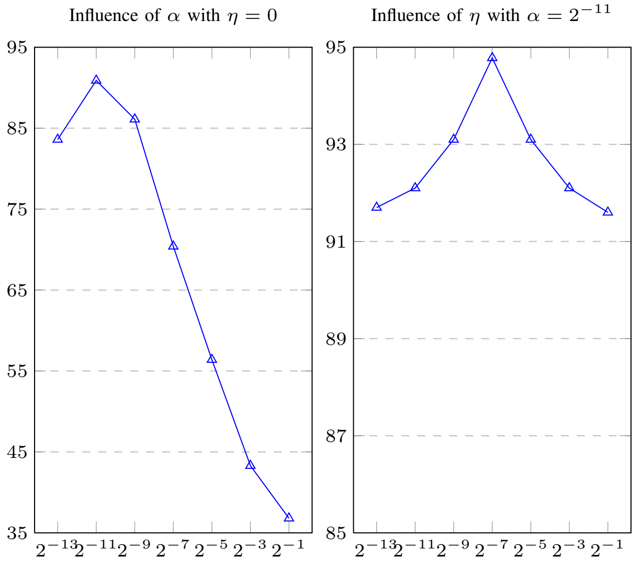

Fig. 3. Influence of α and η on AR database for KCSCDL.

3) AR database: For the AR database, the most appropriate kernel is linear kernel. Optimal parameters $\alpha = 2^{-11}$ and $\eta = 2^{-7}$ are give in Fig.3. The recognition rate reaches 94.78%.

D. Comparison of algorithm performance

In order to show the performance of our algorithm KCSCDL, CSCDL algorithm is also tested on the three databases. The relative data are showed in Table II. From Table II, the comparison proves that KCSCDL algorithm has better performance in face recognition than CSCDL. Both of the two algorithms behave best on the AR database. The data shows the performance of KCSCDL algorithm is capable of increasing the recognition performance after mapping the CSCDL algorithm to the kernel space. KCSCDL outperforms CSDL by 10.74% on the Extended YaleB database and 2.75% on the CMU PIE database. For AR dataset, there is no great effect on improving recognition rate by mapping CSCDL to the kernel space because each algorithm is executed on the linear kernel space which is their optimal kernel .

V. Conclusion

In this paper, motivated by the impressive performance of CSCDL algorithm, the idea that applying it to the kernel space is proposed. The KCSCDL algorithm solves the nonlinear problem and improves the performance of CSCDL further. The block wise coordinate descent algorithm and Lagrange multipliers algorithm are proposed to solve the optimization problem which makes operation more efficient. The experiment results prove that KCSCDL algorithm has superior performance in face recognition to CSCDL algorithm.

ACKNOWLEDGMENT

This paper is supported partly by the National Natural Science Foundation of China (Grant No. 61402535, No. 61271407), the Natural Science Foundation for Youths of Shandong Province, China (Grant No. ZR2014FQ001), Qingdao Science and Technology Project (No. 14-2-4-111-jch), and the Fundamental Research Funds for the Central Universities, China University of Petroleum (East China) (Grant No. 16CX02060A).

REFERENCES

[1] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[2] Robert E Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999.

[3] Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[4] Jeffrey Ho, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. Clustering appearances of objects under varying illumination conditions. In *IEEE conference on Computer vision and pattern recognition*, volume 1, pages I–11. IEEE, 2003.

[5] Dacheng Tao, Xuelong Li, Xindong Wu, and Stephen J Maybank. Geometric mean for subspace selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):260–274, 2009.

[6] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

[7] Bao-Di Liu, Yu-Xiong Wang, Bin Shen, Yu-Jin Zhang, and Martial Hebert. Self-explanatory sparse representation for image classification. In *European Conference on Computer Vision*, pages 600–616. Springer, 2014.

[8] Bao-Di Liu, Yu-Xiong Wang, Yu-Jin Zhang, and Bin Shen. Learning dictionary on manifolds for image classification. *Pattern Recognition*, 46(7):1879–1890, 2013.

[9] Bao-Di Liu, Yu-Xiong Wang, Bin Shen, Xue Li, Yu-Jin Zhang, and Yan-Jiang Wang. Blockwise coordinate descent schemes for efficient and effective dictionary learning. *Neurocomputing*, 178:25–35, 2016.

[10] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[11] Bruno A Olshausen and David J Field. Natural image statistics and efficient coding. *Network: computation in neural systems*, 7(2):333–339, 1996.

[12] Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2443–2446. IEEE, 1999.

[13] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[14] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

[15] Ron Rubinstein, Tomer Peleg, and Michael Elad. Analysis k-svd: a dictionary-learning algorithm for the analysis sparse model. *IEEE Transactions on Signal Processing*, 61(3):661–677, 2013.

[16] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[17] Li Zhang, Wei-Da Zhou, Pei-Chann Chang, Jing Liu, Zhe Yan, Ting Wang, and Fan-Zhang Li. Kernel sparse representation-based classifier. *IEEE Transactions on Signal Processing*, 60(4):1684–1695, 2012.

[18] Jie Zhu, Wan Kou Yang, and Zhen Min Tang. A dictionary learning based kernel sparse representation method for face recognition. *Moshi Shibie Yu Rengong Zhineng/pattern Recognition and Artificial Intelligence*, 25(5):859–864, 2012.

[19] Bao-Di Liu, Bin Shen, Liangke Gui, Yu-Xiong Wang, Xue Li, Fei Yan, and Yan-Jiang Wang. Face recognition using class specific dictionary learning for sparse representation and collaborative representation. *Neurocomputing*, 204:198–210, 2016.

[20] Bao-Di Liu, Liangke Gui, Yuting Wang, Yu-Xiong Wang, Bin Shen, Xue Li, and Yan-Jiang Wang. Class specific centralized dictionary learning for face recognition. *Multimedia Tools and Applications*, pages 1–19, 2015.

[21] Meng Yang, Lei Zhang, Jian Yang, and David Zhang. Metaface learning for sparse representation based face recognition. In *IEEE International Conference on Image Processing*, pages 1601–1604. IEEE, 2010.

[22] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

[23] Bao-Di Liu, Yu-Xiong Wang, Bin Shen, Yu-Jin Zhang, and Yan-Jiang Wang. Blockwise coordinate descent schemes for sparse representation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5267–5271. IEEE, 2014.

[24] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001.

[25] Aleix M Martinez. The ar face database. *CVC Technical Report*, 24, 1998.

[26] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression (pie) database. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 46–51. IEEE, 2002.