Constrained Sparse Concept Coding algorithm with application to image representation

Zhenqiu Shu, Chunxia Zhao, Pu Huang
School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[e-mail: shuzhenqiu@163.com, zhaochx@mail.njust.edu.cn, huangpu3355@163.com]
*Corresponding author: Zhenqiu Shu

Received April 14, 2014; revised June 10, 2014; revised July 7, 2014; accepted August 8, 2014; published September 30, 2014

Abstract

Recently, sparse coding has achieved remarkable success in image representation tasks. In practice, the performance of clustering can be significantly improved if limited label information is incorporated into sparse coding. To this end, in this paper, a novel semi-supervised algorithm, called constrained sparse concept coding (CSCC), is proposed for image representation. CSCC considers limited label information into graph embedding as additional hard constraints, and hence obtains embedding results that are consistent with label information and manifold structure information of the original data. Therefore, CSCC can provide a sparse representation which explicitly utilizes the prior knowledge of the data to improve the discriminative power in clustering. Besides, a kernelized version of our proposed CSCC, namely kernel constrained sparse concept coding (KCSCC), is developed to deal with nonlinear data, which leads to more effective clustering performance. The experimental evaluations on the MNIST, PIE and Yale image sets show the effectiveness of our proposed algorithms.

Keywords: Sparse coding; label information; semi-supervised; constraints; manifold; kernelized

This research was supported by a research grant from the National Natural Science Foundation of China [Gran No. 61272220, 61101197], Jiangsu Province Fund for Graduate Innovation Program [Grant No. CXLX13 19], Natural Science Foundation of Jiangsu Province of China [Grant No. BK2012399] and Project in Jiangsu Key Laboratory of Image and Video Understanding for Social Safety [Grant No. 0920130122006]

http://dx.doi.org/10.3837/tiis.2014.09.015

1. Introduction

In real-world applications, such as face recognition, image retrieval, and data clustering [1, 2, 3, 4, 5, 13, 14, 15], data representation of high-dimensional space is a challenging problem. Generally speaking, the high dimensional data leads to the computational time and memory requirements more expensive. Moreover, traditional methods can perform well in low-dimensional space, but may degrade in high-dimensional space. To solve these issues, many researchers try to seek a representation of the data in a latent semantic "concept" space instead of the original space. Therefore, matrix factorization methods based on different criterions have attracted considerable attention in the last decades.

Principal component analysis (PCA) [2] and linear discriminant analysis (LDA) [3] are the most popular matrix factorization methods. PCA is completely unsupervised learning method, which searches for a projection axis of maximal variance. In contrast with PCA, LDA is a supervised learning method, which aims to seek a transformation that maximizes the between-class scatter and simultaneously minimizes the within-class scatter. The linear methods, however, fail to deal with the nonlinear distribution data. To alleviate this problem, the kernel-based methods are developed to discover the essential structures of nonlinear data. The representative methods are kernel principal component analysis (KPCA) [4] and kernel Fisher discriminant analysis (KFDA) [5], which are the kernel extensions of PCA and LDA, respectively. Extensive experiments have shown the effectiveness of KPCA and KFDA in many real-world applications.

A common problem of the previously mentioned methods is that they fail to discover the underlying intrinsic manifold structure. To overcome this deficiency, manifold-based learning methods are straightforward in detecting the nonlinear structures, which have been of wide concern. Yan *et al*. [6] proposed a general framework, called graph embedding, for dimensionality reduction. Many manifold-based learning methods, such as isometric feature mapping (ISOMAP) [7], locally linear embedding (LLE) [8], laplacian eigenmaps (LE) [9] and locality preserving projection (LPP) [10] can be integrated into this framework. To consider the label information of labeled samples, He *et al*. [11] presented a novel semi-supervised learning algorithm, called constrained graph embedding (CGE) for feature extraction and data representation. CGE incorporates the limited label information into graph embedding as additional constraints. Experimental results on real data sets have illustrated the effectiveness of CGE.

Different from all the aforementioned methods, there is psychological and physiological evidence for parts-based representation in the cognitive process of human brain. One of the well-known parts-based methods is non-negative matrix factorization (NMF) [12] that tries to decompose the original date matrix into the product of two non-negative matrices. In particular, due to the non-negative limitation, NMF allows only additive, not subtractive, operation, which leads to a parts-based representation of the original data. Concept factorization (CF) [14] is a variant of NMF in that each cluster is linearly represented by a few data, and each data is linearly represented by the cluster centers. The major advantage of CF over NMF is that it can be performed on positive as well as negative data and simultaneously kernelized to further improve performance. Until recently, massive other works have been done [16, 17, 18, 19, 20, 21, 22] on extensions of NMF.

However, the coefficient matrix of the above methods is usually dense. This is contrary to our understanding that the coefficient matrix is sparse since each sample is represented by a

linear combination of only a few concepts. Inspired by biological visual systems, sparse coding (SC) is recently proposed for data representation and has been widely applied in many fields [23, 24, 25, 26, 27]. Traditional SC algorithm, however, fails to take consideration of the geometrical structure information, which can significantly improve the discriminant ability in real-world applications [16, 17, 28, 29]. Consequently, a variety of extensions of SC have been developed to explore the geometrical structure of the data by adding some constraints. Wang *et al*. [30] proposed a novel technique, called locality-constrained linear coding (LLC), for image representation and classification. LLC preserves the local geometric structure in feature coding process. Mairalet *et al*. [31] developed simultaneous sparse coding as a framework where groups of similar signals are jointly decomposed by adding group sparsity regularization term. Gao *et al*. [32] presented a laplacian sparse coding (LSC) framework to solve the data classification and tagging problems. LSC incorporates the laplacian regularization into the mode of SC to preserve the consistence of similar local features. Similar to LSC, Zhang *et al*. [33] proposed a graph regularization sparse coding (GSC) approach for image representation. GSC captures the intrinsic manifold structure with resort to laplacian graph regularization. Experimental results on image databases have shown the effectiveness of GSC. However, one evident drawback of all the aforementioned SC methods is computationally expensive to optimize their models. To overcome this limitation, Cai *et al*. [34] proposed a very efficient algorithm, namely sparse concept coding (SCC), for visual analysis. One of the major advantages of SCC is very efficient because it only solves a sparse eigenvalue problem and two regression problems. For each sample, SCC seeks a sparse representation of basis vectors that are embedded the semantic structure information of the data.

Unfortunately, SCC is completely unsupervised without regard to label information. Some previous research efforts reveal that the simultaneous use of the labeled data and unlabeled data can further improve performance in clustering [35, 36]. Thus, we propose a novel semi-supervised learning algorithm, called *constrained sparse concept coding (CSCC)*, for data representation. CSCC considers limited label information and the intrinsic manifold structure of data, simultaneously. Our empirical study on benchmark data sets shows the promising results of our proposed algorithm. The contributions of this paper are as follows:

(1) Our proposed CSCC preserves some merits of SCC. For example, it exploits the manifold structure of the data, and simultaneously only solves a sparse eigenvalue problem and two regression problems. Therefore, CSCC is also computationally efficient in comparison with other sparse coding methods.

(2) Compared with SCC, CSCC is a semi-supervised learning algorithm, and thus considers limited label information as additional hard constraints. Moreover, CSCC incorporates the label information into graph embedding in a parameter-free manner. As a result, CSCC not only respects the intrinsic manifold structure of the data, but also takes advantage of the label information of the labeled data.

(3) We also propose another algorithm, called *kernel constrained sparse concept coding* (KCSCC), based on our proposed CSCC. With the kernel trick, nonlinear relationships among data are transformed into linear relationships in the high-dimensional kernel space. Therefore, we may obtain more effective performance in most cases.

The remainder of this paper is organized as follows: Section 2 gives a brief review of the related work. Section 3 introduces our proposed algorithms. Section 4 provides the experimental results to demonstrate the effectiveness of the proposed algorithms. Finally, we provide some concluding remarks and suggestions for future work in Section 5.

2. Related Work

In the past few years, sparse coding is proposed based on matrix factorization for data representation, which aims to seek a sparse linear combination of basis vectors for each sample. Therefore, the model of SC can be defined based on matrix factorization model by imposing sparse constraints on representation coefficient.

Given a sample set $\mathbf{X}=[\mathbf{x}_1,\cdots,\mathbf{x}_n]\in\Re^{m\times n}$, where x_i stands for a sample. Let $\mathbf{U}=[\mathbf{u}_1,\cdots,\mathbf{u}_k]\in\Re^{m\times k}$ be the basis matrix, where u_i can be regarded as a basis in the new representation space. Let $\mathbf{A}=[\mathbf{a}_1,\cdots,\mathbf{a}_n]\in\Re^{k\times n}$ be the coefficient matrix, where a_i denotes the representation coefficient of x_i. Thus, the objective function of SC can be formally expressed as:

$$\min_{U,A}\|X-UA\|_F^2+\beta|A|_0 \tag{1}$$

where $|A|_0$ denotes the $\ell 0$-norm and enforces the sparsity on A, and β is a regularization parameter.

Unfortunately, $\ell 0$-norm minimization problem is not convex. Therefore, finding the sparsest solution of Eq. (2) is a NP-hard problem and computationally expensive. Recent studies have shown that if the solution is sparse enough, the sparsest solution of $\ell 1$-norm minimization problem is equal to the solution of $\ell 0$-norm minimization problem [37, 38]. Thus, an alternative formulation for Eq. (1) is to replace $\ell 0$-norm regularization by $\ell 1$-norm regularization to enforce sparsity constraints:

$$\min_{U,A}\|X-UA\|_F^2+\beta\|A\|_1 \tag{2}$$

where $|A|_1$ denotes $\ell 1$-norm of the coefficient matrix A. Here, we can employ standard linear programming methods to solve the $\ell 1$-norm minimization problem in Eq. (2).

3. Our Proposed Methods

3.1 Matrix Factorization

Generally, factorization of matrices may be non-unique, and hence varieties of matrix factorization methods have been developed by imposing different constraints. Specifically, given a data set $\mathbf{X}=[\mathbf{x}_1,\cdots,\mathbf{x}_n]\in\Re^{m\times n}$, matrix factorization method tries to seek two matrices $\mathbf{U}\in\Re^{m\times k}$ and $\mathbf{A}\in\Re^{k\times n}$ satisfying:

$$X=UA$$

where U denotes the basis vectors and A is the coefficient matrix of the samples under the basis vectors U. Thus, the objective function of matrix factorization can be formalized as:

$$\min_{U,A}\|X-UA\|_F^2 \tag{3}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm of a matrix.

The models of various matrix factorization methods, such as PCA, LDA, Graph Embedding, NMF, CF, and SC can be constructed by adding different constraints on Eq. (3) based on different purposes. In this way, we can impose some constraints on Eq. (3) that the basis vectors U should be embedded the semantic structure of the data, and simultaneously the

representation coefficient *A* should be sparse. In the next subsection, we first introduce the CSCC algorithm in detail.

3.2 CSCC algorithm

CSCC is a three-step algorithm for data representation. The first step is *concept extraction*. CSCC takes the label information as additional constraints into graph embedding. Therefore, the low dimensional *concepts* exploit both limited label information and the manifold structure information. The second step is *basis learning*. CSCC aims to learn a basis that can best fit the concepts. As a result, the basis are encoded the semantic structure of the original data. The final step is *sparse representation learning*. We can employ LARs [39] algorithm to learn a sparse representation for each data.

3.2.1 Concept extraction

Given a data set $\mathbf{X}=[\mathbf{x}_1,\cdots,\mathbf{x}_n]\in\Re^{m\times n}$, an adjacency graph $G=\{X,W\}$ can be constructed with data set *X*, where *W* denotes a weighted matrix. The elements of the weighted matrix *W* can be usually defined as:

$$W_{\mathrm{ij}}=\begin{cases}1 & \text{if } x_i\in \mathrm{N}_p(x_j) \text{ or } x_j\in \mathrm{N}_p(x_i)\\ 0 & \text{otherwise}\end{cases}$$

where $\mathrm{N}_p(x_j)$ is the set of *p* nearest neighbors of x_j, *L=D-W* is the Laplacian matrix, *D* is a diagonal matrix and $D_{ii}=\sum_j W_{ij}$. Let $y=[y_1,\cdots,y_n]^T$ be the map from the graph to the real line. According to the reference [13], we will introduce how to incorporate label information of labeled data into graph embedding.

Assume that the first *l* samples $x_1,\cdots,x_l$ belong to *c* classes as labeled set and the rest *n-l* samples $x_{l+1},\cdots,x_n$ are unlabeled. We first construct an indicator matrix $\mathbf{M}\in\Re^{l\times c}$ where $m_{ij}=1$ if x_i is labeled with the *j*th class; $m_{ij}=0$ otherwise. Once obtaining the indicator matrix *M*, we define the label constraint matrix $\mathbf{S}\in\Re^{n\times(n-l+c)}$ as follows:

$$S=\begin{pmatrix}M_{l\times c} & 0\\ 0 & I_{n-l}\end{pmatrix} \tag{4}$$

where I_{n-l} is a $(n-l)\times(n-l)$ identity matrix. For example, given *n* samples among which x_1 is from the first class, x_2, x_3 and x_4 are from the second class, x_5 and x_6 are from the third class, and the rest *n*-6 samples are unlabeled. Thus, the label constraint matrix *S* can be defined:

$$S=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & I_{n-6}\end{pmatrix}$$

where I_{n-6} denotes a $(n-6)\times(n-6)$ identity matrix. Note that concept extraction can map each image x_i to y_i from the original feature space to the new concept space. To take advantage of limited label information, we can impose the label constraints by introducing an auxiliary matrix *z*:

$$y = Sz \tag{5}$$

From the Eq. (5), it can be found that if x_i and x_j share the same label, then $y_i = y_j$. Thus, we have:

$$\sum_{i,j=1}^{n}(y_i - y_j)^2 W_{ij} = y^T L y = z^T S^T L S z$$

And

$$y^T D y = z^T S^T D S z$$

Thus, the minimization problem reduces to:

$$\begin{aligned} &\max\ z^T S^T L S z \\ &s.t.\ z^T S^T D S z = 1 \end{aligned} \tag{6}$$

We can obtain the optimal vector *z* of Eq. (6) by solving the following generalized eigenvalue problem:

$$S^T L S z = \lambda S^T D S z \tag{7}$$

Let $Z = [z_1, \cdots, z_d]$, z_i's are the eigenvectors of the generalized eigenvalue problem in Eq. (7) corresponding to the smallest eigenvalue. After we obtain *z*, the *y* can be derived by Eq. (5). Let $Y = [y_1, \cdots, y_d]$, each row of *Y* is called a "*concept*" which embeds the semantic structure information for each sample. If there is no labeled data, we can obtain $S = I_n$. In this case, the CSCC method reduces to SCC method.

3.2.2 Basis learning

Considering the label information and the manifold structure at the same time, we aim to find a basis *U* that can best fit *Y*. That is to say, the basis *U* needs to satisfy $X^T U = Y$. Unfortunately, the system is under-determined, such *U* does not exist. A feasible way to solve this problem is to impose a penalty on the norm of *U* as follows:

$$\min_U \left\| Y - X^T U \right\|^2 + \alpha \left\| U \right\|^2 \tag{8}$$

where α is the nonnegative constant parameter and $\alpha\left\|U\right\|^2$ can avoid over-fitting. In statistical learning, the model in Eq. (8) is called *Ridge Regression* problem [41].

By taking the derivative of Eq. (8) with respect to *U* and setting it to zero, the optimal solution U^* of Eq. (8) can be expressed as follows:

$$U^* = (XX^T + \alpha I)^{-1} XY \tag{9}$$

Actually, the dimensionality of the images is so high that it is extremely time-consuming to solve $(XX^T + \alpha I)^{-1}$. Fortunately, some iterative algorithms, such as LSQR [40], are used to directly find the solution of the regression problem in Eq. (8).

3.2.3 Sparse Representation Learning

Suppose that a_i and x_i denote the *i*th column vector of *A* and *X*, respectively. Once we obtain the basis *U*, the coefficient a_i of the sample x_i can be solved through the following minimization problem:

$$\min_{a_i} \|x_i - Ua_i\|^2 + \beta|a_i| \tag{10}$$

where $|a_i|$ indicates the $\ell 1$-norm of a_i and $\beta > 0$ is a constant parameter. The $\ell 1$-norm minimization problem in Eq. (10) is called *LASSO* in statistical learning [41].

Note that the minimization optimization problem in Eq. (10) can be reformulated as follows:

$$\begin{aligned} &\min_{a_i} \|x_i - Ua_i\|^2 \\ &s.t. \quad |a_i| \le \gamma \end{aligned} \tag{11}$$

Fortunately, the Least Angel Regression (LARs) [39] algorithm can be used to solve the minimization optimization problem in (11). Thus, we need to set the cardinality (the number of non-zero entries) of a_i without the parameter γ. In this way, it is easy to control the sparseness of representation coefficient a_i.

Algorithm 1: CSCC algorithm

Input: Given a set of *n* samples $X = [x_1, \cdots x_l, x_{l+1}, \cdots, x_n] \in \mathbb{R}^{m \times n}$, $X_L = \{x_i\}_1^l$ are labeled and $X_U = \{x_i\}_{l+1}^n$ are unlabeled.

Output: The coefficient matrix *A*.

1:construct the label constraint matrix *S* by Eq. (4);

2:compute the eigenvectors *Z* of the generalized eigenvalue problem in Eq. (7) corresponding to the *d* smallest eigenvalue and then obtain *Y* by Eq. (5);

3:calculate the basis matrix U^* in Eq. (8) by the LSQR algorithm;

4:compute the coefficient vector a_i in Eq. (11) by LARs algorithm. Thus, the coefficient matrix is given by: $A = [a_1, \cdots, a_n]$.

3.2.4 Computational Complexity of CSCC

Let *q* be the average number of non-zero entries in each data, and $q \le m$. *p* denotes the size of nearest neighbors, and *d* represents the dimensionality of the concept. The total computational complexity of CSCC algorithm includes three parts:

(1) In concept extraction phase, we need $O(n^2q + n^2p)$ operations to calculate the nearest neighbor graph and $O(dnp + 2n)$ operations to calculate *Y*. Considering $d << n$, we need $O(n^2q + n^2p)$ operations to extract the concepts.

(2) In basis learning phase, the time complexity of calculating the U^* in Eq. (9) with LSQR method is $O(dnq)$.

(3) In sparse representation phase, the time complexity of calculating the coefficient matrix A in Eq. (11) with LARs method is $O(d^3+md^2)$.

In summary, the total complexity of our CSCC algorithm is $O(n^2q+n^2p+d^3+md^2)$. From the reference [34], it is easy to check that the total computational cost of CSCC is equal to that of SCC.

3.3 KCSCC algorithm

To our knowledge, the kernel trick is proposed in SVM to handle classification tasks that cannot be linearly separable in the original space. Once the samples in original input space are mapped to the high dimensional feature space via kernel trick, the linear algorithms in pattern recognition can be employed to handle the data in kernel space.

Motivated by the fact that kernel methods can deal with the nonlinear structure data, we propose a nonlinear extension of our CSCC, namely KCSCC, which seeks a sparse representation of nonlinear data. Similar to CSCC, our proposed KCSCC is also a three-step method including concept extraction, basis learning and sparse representation learning.

The concept extraction phase of our proposed KCSCC is the same as that of CSCC. Consequently, we only introduce the basis learning and sparse representation learning of KCSCC.

3.3.1 Basis learning

Suppose that there exists a nonlinear mapping function $\phi:\Re^n\to\mathbb{F}$ which can map the original feature to the kernel feature space:

$$x\xrightarrow{\phi}\phi(x)$$

Meanwhile, define the $K(\cdot,\cdot)$ as a Gram matrix, with elements

$$K_{ij}=\kappa(\varphi(x_i),\varphi(x_j)) \tag{12}$$

where $\kappa(\cdot,\cdot)$ is a positive semi-definite kernel function.

Similar to basis learning phase of CSCC, KCSCC needs to find a basis $\overline{U}$ in high dimensional space to satisfy:

$$K\overline{U}=Y \tag{13}$$

where Y stands for embedding results in Eq. (4). In fact, the system in Eq. (13) is also under-determined, such $\overline{U}$ does not exist. A natural approach is to approximate $\overline{U}$ by solving:

$$(K+\alpha I)\overline{U}=Y \tag{14}$$

where I denotes an identity matrix and $\alpha>0$ is a parameter. Thus, the optimal solution $\overline{U}^*$ of Eq. (14) is expressed as follows:

$$\overline{U}^*=(K+\alpha I)^{-1}Y \tag{15}$$

Similarly, it is computational expensive to solve $(K+\alpha I)^{-1}$. Fortunately, finding the solution of Eq. (14) can be converted into solving a regression problem. First, we need to define a projective function in the kernel space as follows:

$$f(x)=\langle\overline{U},\phi(x)\rangle=\sum_{i=1}^{n}\bar{u}_iK(x,x_i) \tag{16}$$

It is easy to show that the optimal solution $\overline{U}^*$ of Eq. (14) is the same as that of the following regularized regression problem:

$$\min_{f \in F} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \alpha \| f \|_{\kappa}^2 \tag{17}$$

where $f(x_i)$ and y_i are the *i*th element of $f(x)$ and *Y*, respectively; F is the RKHS associated with Mercer kernel κ and $\| \ \|_{\kappa}$ is the corresponding norm.

Similarly, the regression problem in Eq. (17) can be directly solved by using the LSQR [40] algorithm.

3.3.2 Sparse representation learning

Suppose that $\overline{a}_i$ and x_i are the *i*th column of $\overline{A}$ and *X*, respectively. After we get the basis $\overline{U}$, the coefficient $\overline{a}_i$ of the sample x_i can be computed as follows:

$$\min_{\overline{a}_i} \left\| K(:, x_i) - \overline{U}\overline{a}_i \right\|^2 + \beta \left| \overline{a}_i \right| \tag{18}$$

where $\left| \overline{a}_i \right|$ enforces the sparsity on $\overline{a}_i$, $K(:, x_i) = [K(x_1, x_i), \cdots, K(x_n, x_i)]^T$ and $\beta > 0$ is a regularization parameter.

Note that the $\ell 1$-norm minimization problem in Eq. (18) can be reformulated as:

$$\begin{aligned} &\min_{\overline{a}_i} \left\| K(:, x_i) - \overline{U}\overline{a}_i \right\|^2 \\ &s.t. \quad \left| \overline{a}_i \right| \le \gamma \end{aligned} \tag{19}$$

Similarly, we can employ LARs [39] algorithm to directly solve the Eq. (19). Therefore, we need to specify the cardinality of $\overline{a}_i$ without the parameter γ.

Algorithm 2: KCSCC algorithm

Input: Given a set of *n* samples $X = [x_1, \cdots x_l, x_{l+1}, \cdots, x_n] \in \mathbb{R}^{m \times n}$, $X_L = \{x_i\}_1^l$ are labeled and $X_U = \{x_i\}_{l+1}^n$ are unlabeled.

Output: The coefficient matrix $\overline{A}$.

1:construct the label constraint matrix *S* by Eq. (4);

2:compute the eigenvectors *Z* of the generalized eigenvalue problem in Eq. (7) corresponding to the *d* smallest eigenvalues and then obtain *Y* by Eq. (5);

3:map the original data to the kernel feature space by Eq. (16), and then compute the basis vectors $\overline{U}^*$ in Eq. (17) by the LSQR algorithm;

4:compute the representation coefficient $\overline{a}_i$ in Eq. (19) by LARs algorithm. Thus, the coefficient matrix is given by: $\overline{A} = [\overline{a}_1, \cdots, \overline{a}_n]$.

3.3.3 Computational complexity of KCSCC

Similar to CSCC, the consuming time of the concept extraction phase is $O(n^2 q + n^2 p)$ in our proposed KCSCC.

In basis learning phase, we need $O(mn^2)$ operations to map the original data to the kernelspace in Eq. (12). In addition, we also need $O(n^3+n^2d)$ to solve the basis vectors by using the LSQR algorithm. Considering $d<<n$, the total complexity of this phase is $O(mn^2+n^3)$.

In sparse representation learning phase, we need $O(mn^2)$ operations to map the original data to the kernel space. Besides, we need $O(d^3+md^2)$ operations to obtain the coefficient matrix $\overline{A}$ by LARs algorithm. This total complexity of sparse representation learning can be written as $O(mn^2+d^3)$.

In summary, the total complexity of our proposed KCSCC is $O(n^2q+n^2p+n^3+mn^2)$.

4. Experimental Classification Results and Analysis

Recent studies have shown that matrix factorization methods are very powerful in clustering tasks. We carry out several experiments to evaluate the performance of our proposed algorithms on MNIST, PIE and Yale image databases. In our experiments, the two evaluation metrics of the clustering involve accuracy (AC) and normalized mutual information (NMI). The detail definitions of AC and NMI are found in [14].

4.1 Performance Evaluation and Comparisons

In this subsection, we will systematically conduct the evaluations on some image datasets and compare the performances of CSCC and KCSCC with some other algorithms such as K-means, CF, NMF, PCA and SCC.

In these experiments, we randomly choose $K(=3, 4,\cdots,10)$ categories from the image data sets for clustering. For our proposed semi-supervised algorithm (CSCC and KCSCC), we randomly pick up some date samples for each subject to provide the available label information, and other data samples are unlabeled. For each given cluster number *K*, the experiment process is repeated 10 times and the average performance is recorded as the final result.

4.1.1 Experiments on MNIST handwritten database

The MNIST handwritten database includes a training set of 60 000 examples, and a test set of 10 000 examples. In this experiment, 500 samples are selected for clustering. These digit images have been normalized to 28×28 gray scale images. **Fig. 1** shows some handwritten samples from MNIST database.

Fig. 1. Images from the MNIST database

In this experiment, we randomly pick up 50 samples for each subject and randomly select 20% from these samples as available label information. The rest digit images are left as unlabeled data. This is, for each subject, the labeled data are 10 images and the rest unlabeled

data are 40 images, which are mixed as a whole for clustering. The experimental results are summarized in **Table 1**. It is noticeable that our proposed methods consistently outperform other competitors. As shown in **Table 1**, compared with the best algorithm, i.e., SCC, CSCC and KCSCC achieve 6.6 % and 9% improvement in AC, respectively. For NMI, CSCC and KCSCC algorithms achieve 7.3% and 7.6% improvement, respectively.

Table 1. Clustering performance on MNIST database

(a)AC

K	K-means	CF	NMF	PCA	SCC	CSCC	KCSCC
3	0.863	0.631	0.721	0.833	0.752	0.867	**0.869**
4	0.795	0.685	0.661	0.724	0.672	0.862	**0.871**
5	0.714	0.600	0.653	0.701	0.672	0.735	**0.744**
6	0.674	0.595	0.644	0.649	0.671	0.709	**0.733**
7	0.602	0.574	0.583	0.584	0.629	0.674	**0.679**
8	0.593	0.575	0.564	0.583	0.651	0.704	**0.729**
9	0.579	0.579	0.559	0.563	0.618	**0.729**	0.716
10	0.540	0.654	0.508	0.574	0.646	0.660	**0.690**
avg	0.670	0.612	0.612	0.651	0.664	0.743	**0.754**

(b)NMI

K	K-means	CF	NMF	PCA	SCC	CSCC	KCSCC
3	0.661	0.373	0.479	0.650	0.555	0.708	**0.709**
4	0.569	0.388	0.416	0.518	0.453	0.687	**0.701**
5	0.597	0.529	0.516	0.573	0.565	0.621	**0.632**
6	0.579	0.512	0.543	0.571	0.598	**0.644**	0.618
7	0.535	0.489	0.500	0.519	0.580	0.601	**0.613**
8	0.546	0.508	0.502	0.512	0.606	0.647	**0.657**
9	0.557	0.526	0.501	0.543	0.593	**0.631**	0.620
10	0.535	0.563	0.525	0.549	0.614	0.616	**0.624**
avg	0.572	0.486	0.498	0.554	0.571	0.644	**0.647**

4.1.2 Experiments on PIE face database

The PIE face database consists of 41,368 images of 68 individuals. The face images were captured by 13 synchronized cameras and 21 flashes under different pose, illumination and expression. In this experiment, we choose the frontal pose (C27) with varying lighting and illumination which leave us about 46 images per subject. The gray face images are normalized to 32×32 pixels. Some sample images from the PIE database are shown in **Fig. 2**.

Fig. 2. Face examples from the PIE database

In this experiment, we randomly choose 46 samples per subject for clustering. For each category, we randomly pick up 9 face images to provide label information as labeled set, the rest images without label information as unlabeled set. Then the labeled set and unlabeled set are mixed for clustering. Shown in **Table 2** are the comparison results of all matrix factorization methods. As we can see, KCSCC achieves the highest average AC 68.7% and the highest average NMI 67.5%. The average AC of CSCC achieves nearly 3.7% improvement and the average NMI achieves 6.2% improvement over the SCC, respectively. Compared to the CSCC, the average AC of KCSCC is better 1.8% and the average NMI is slightly better 1.1%.

Table 2 Clustering performance on PIE database

(a)AC

K	K-means	CF	NMF	PCA	SCC	CSCC	KCSCC
3	0.511	0.556	0.530	0.500	0.733	**0.784**	0.761
4	0.372	0.438	0.490	0.398	0.643	0.678	**0.728**
5	0.473	0.460	0.503	0.427	0.706	0.714	**0.736**
6	0.355	0.364	0.418	0.353	0.603	0.666	**0.725**
7	0.269	0.319	0.444	0.321	0.644	0.678	**0.702**
8	0.271	0.264	0.397	0.273	0.599	**0.631**	0.604
9	0.278	0.279	0.418	0.227	0.573	0.616	**0.663**
10	0.232	0.240	0.357	0.245	0.553	**0.581**	0.576
avg	0.345	0.365	0.445	0.343	0.632	0.669	**0.687**

(b) NMI

K	K-means	CF	NMF	PCA	SCC	CSCC	KCSCC
3	0.189	0.299	0.212	0.206	0.533	0.634	**0.653**
4	0.121	0.182	0.246	0.142	0.519	0.594	**0.651**
5	0.360	0.302	0.391	0.332	0.685	0.691	**0.749**
6	0.192	0.199	0.269	0.203	0.548	**0.677**	0.666
7	0.200	0.192	0.374	0.236	0.671	0.687	**0.707**
8	0.194	0.144	0.353	0.197	0.632	**0.666**	0.640
9	0.218	0.185	0.389	0.178	0.619	0.699	**0.707**
10	0.186	0.144	0.354	0.200	0.609	**0.662**	0.627
avg	0.208	0.206	0.324	0.212	0.602	0.664	**0.675**

4.1.3 Experiments on Yale database

The Yale face database contains 15 subjects each providing 11 different images, thus 165 face images in total. For some subjects, the face images were taken under various lighting conditions and facial expressions. All face images have been normalized to 32×32 pixels. **Fig. 3** shows some face images of two subjects from the Yale database.

Fig. 3. Face examples from the Yale database

In this experiment, we randomly pick up the 20% images as labeled data, the rest images as unlabeled data for each category. In other words, since each category includes 11 images, we randomly choose 2 images to provide the label information as additional constraints, and consider the rest 9 images as the unlabeled data. Finally, the labeled data and unlabeled data are mixed as a whole for clustering. The experimental results are summarized in **Table 3**. In general, our proposed methods are always better than other algorithms on Yale database. Compared with the SCC algorithm, CSCC and KCSCC algorithms achieve 3.9% and 4.5% improvement in AC, respectively. For NMI, CSCC and KCSCC achieve 4.2% and 5.6% improvement, respectively. Compared with the best NMF algorithm, CSCC and KCSCC algorithms achieve 2.7% and 3.3% improvement in AC, respectively. For NMI, CSCC and KCSCC algorithm achieve 3.5% and 4.9% improvement, respectively.

Table 3. Clustering performance on Yale database

(a)AC

K	**K-means**	**CF**	**NMF**	**PCA**	**SCC**	**CSCC**	**KCSCC**
3	0.654	0.672	0.715	0.667	0.739	0.776	**0.779**
4	0.522	0.509	0.622	0.609	0.595	**0.672**	0.663
5	0.436	0.418	0.529	0.524	0.530	0.537	**0.556**
6	0.472	0.424	0.463	0.460	0.474	0.488	**0.506**
7	0.467	0.431	0.490	0.471	0.483	0.499	**0.519**
8	0.443	0.404	0.479	0.468	0.450	**0.507**	0.498
9	0.440	0.420	0.460	0.461	0.434	0.479	**0.499**
10	0.427	0.356	0.449	0.457	0.410	0.465	0.455
avg	0.483	0.454	0.526	0.515	0.514	0.553	**0.559**

(b) NMI

K	**K-means**	**CF**	**NMF**	**PCA**	**SCC**	**CSCC**	**KCSCC**
3	0.435	0.449	0.486	0.471	0.507	0.560	**0.573**
4	0.341	0.313	0.430	0.422	0.418	**0.492**	0.475
5	0.302	0.263	0.360	0.332	0.354	0.384	**0.392**
6	0.371	0.288	0.354	0.351	0.373	0.385	**0.400**
7	0.401	0.338	0.394	0.393	0.405	0.414	**0.444**
8	0.405	0.340	0.416	0.401	0.384	0.442	**0.454**
9	0.426	0.384	0.434	0.434	0.418	0.459	**0.492**
10	0.419	0.343	0.422	0.452	0.377	0.442	**0.458**
avg	0.388	0.340	0.412	0.407	0.405	0.447	**0.461**

4.2 Parameters Discussion

In our experiments, the size of neighborhood p and the regularization parameter α are empirically set to 5 and 0.1, respectively. We implement our proposed KCSCC algorithm with degree 2 polynomial kernel. Suppose that the cardinality denotes the non-zero number of

representation coefficient and is empirically specified as half of the number of the basis. In addition, the number of basis vectors is empirically set as the number of clusters.

Fig. 4 shows how the performances of our proposed algorithms vary with the cardinality parameter on all image databases. In each experiment, we simply use the first 10 categories samples for clustering. Since the number of basis vectors is empirically set to the number of clusters, there are 10 basis vectors in new concept space. In other words, we can use a 10 dimension vector to represent each data on concept space, where the vector is generally called representation coefficient. From **Fig. 4** we can see that each high dimensional data, such as 1024 dimensionality, can be represented by the coefficient with only 3 non-zero entries in new concept space. Therefore, it indicates that the representation for each image is very sparse in concept space. This observation suggests that we can use a linear combination of only a few concepts to represent each image. This is consistent with our common knowledge since most of the images contain only a few concepts.

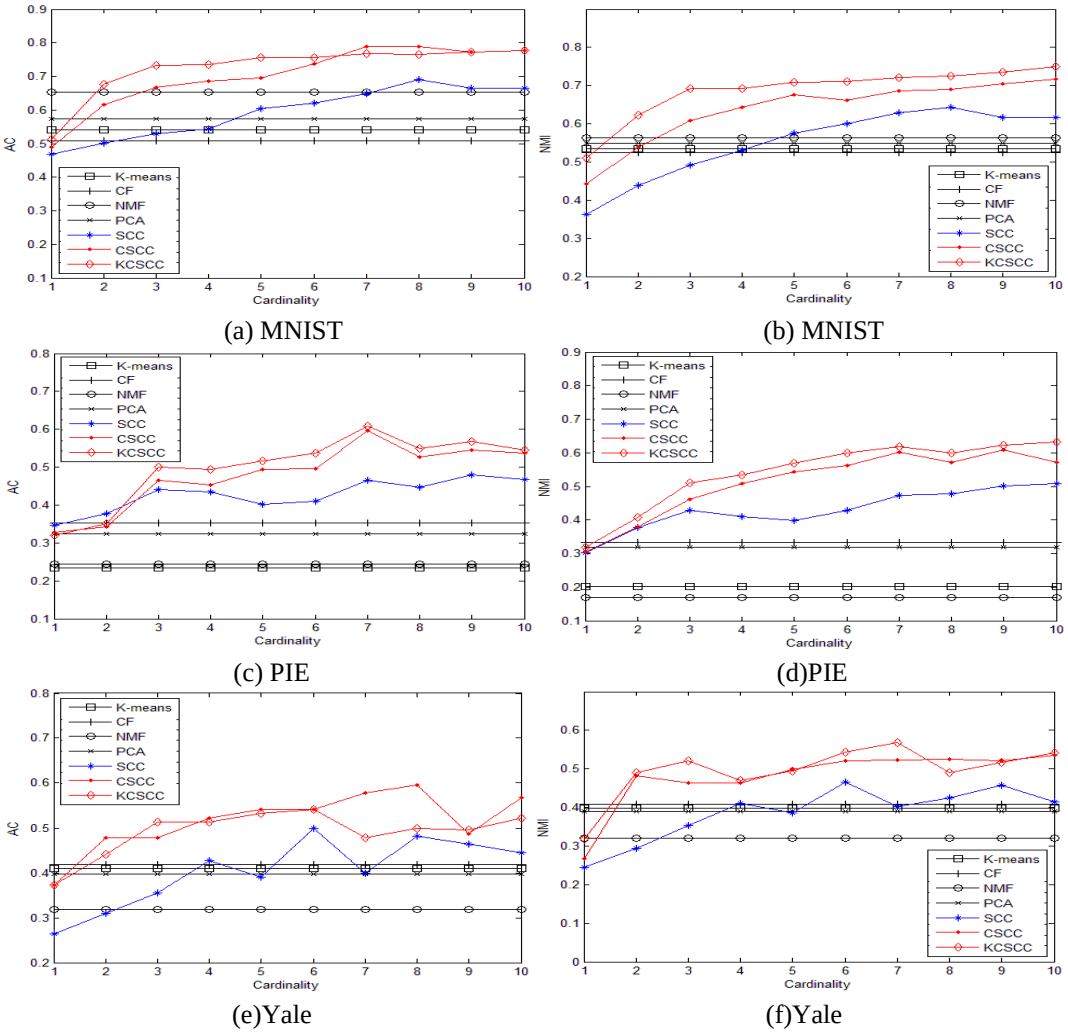

Fig. 4. The performance with varied the cardinality

Fig. 5 shows the performances of our proposed algorithms with the increasing of the number of labeled data. K-means, CF, NMF, PCA and SCC are unsupervised learning

algorithms without regard to label information of the data. CSCC and KCSCC, however, are semi-supervised learning algorithms, and thus the performances have a close relation to the number of the labeled data. In each experiment, we randomly choose 10 categories from image database and randomly pick up different ratios of data samples per class as labeled data for clustering. Then 10 independent experiments are taken to calculate the average performance and the results are shown in **Fig. 5**. We carry out the experiments with the ratio of the labeled data ranging from 10% to 50%. As can be seen in **Fig. 5**, CSCC and KCSCC outperform other unsupervised learning algorithms. Moreover, it can be found that the performances of our semi-supervised learning algorithms, such as CSCC and KCSCC, become better as the number of labeled data increases. It implies that label information plays an important role for image representation. This is also consistent with our understanding for the role of label information in semi-supervised learning algorithm.

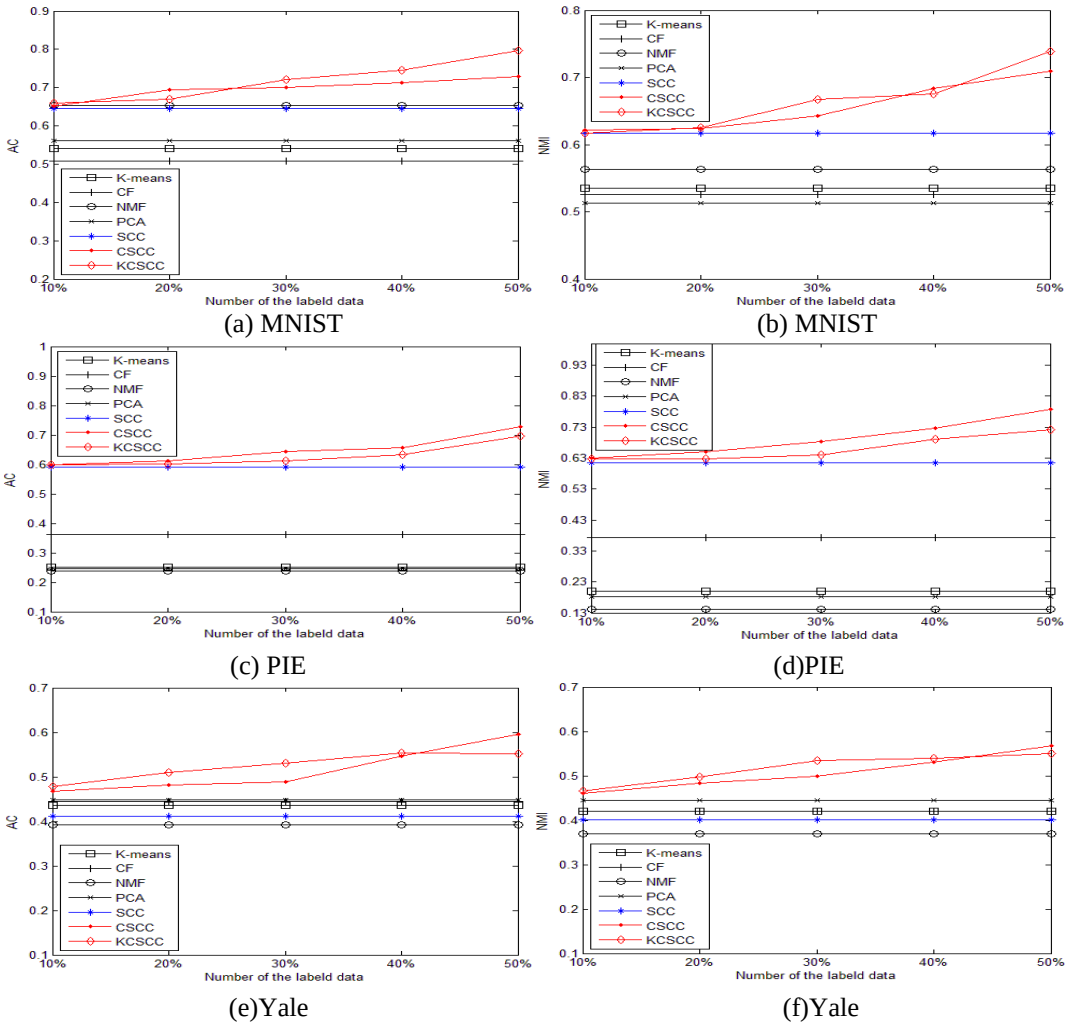

Fig. 5. The performance with varied the number of labeled data

Fig. 6 displays the results of all comparative algorithms with varied the number of basis vectors on three image datasets. For various matrix factorization algorithms, the selection of the number of basis vectors is a fundamental topic. Throughout above experiments, the

number of basis vectors is empirically set to the number of the clusters. In this experiment, we randomly choose 10 categories data from the image database for clustering and repeat the experiment process 10 times. Then the average performances of all methods are shown in **Fig. 6**. As we can see, when the number of basis vectors ranges from 5 to 250, it is easy to check that the performances of CSCC and KCSCC are superior to other matrix factorization algorithms on all databases. It can be found that our proposed algorithms achieve the highest performance when the number of basis vectors is equal to the number of classes. Besides, we can see that the performances of CSCC and KCSCC are relatively stable with varied the number of basis vectors.

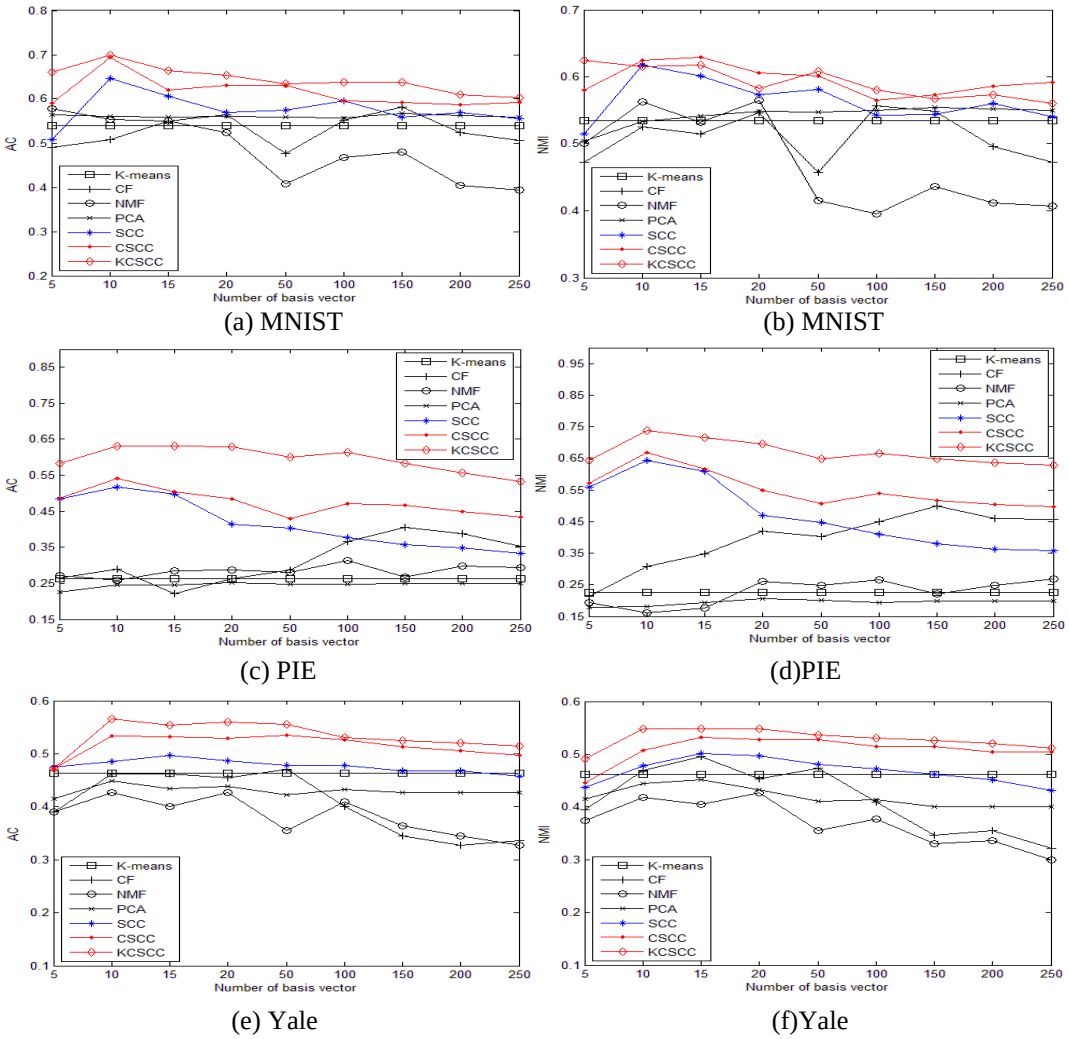

Fig. 6. The performance with varied the number of basis vectors

4.3 Observations

These experiments on MNIST, PIE and Yale image data sets reveal some interesting points: (1) As we can see, SCC is superior to K-means, NMF, CF and PCA on MNIST and PIE databases. Unfortunately, the performance of SCC is inferior to NMF and PCA on Yale

database. Thus, we can see that SCC cannot achieve the best performance compared with NMF and PCA in all data sets. The reason may be that SCC only explores the manifold structure of the data.
(2) CSCC can obtain the better performance than SCC on all image databases. The reason is that SCC is completely unsupervised. Our proposed CSCC, however, is a semi-supervised learning algorithm, and thus incorporates limited label information into graph embedding. The embedding results of CSCC are consistent with the prior knowledge such that the images from the same class are merged together and simultaneously preserve the manifold structure information of data. The experimental results demonstrate that limited label information, when used in conjunction with geometric manifold structure information, can improve the performance in clustering.
(3) Regardless of the database, we can see that the average performance of KCSCC consistently outperforms CSCC. The main reason is that KCSCC not only preserves the advantages of CSCC, but also can handle the nonlinear structure data via kernel trick. Therefore, KCSS provides more discriminative power than other competitors, such as SCC and CSCC.
(4) As we can see, when the minimum for cardinality is 3, our proposed CSCC and KCSCC algorithms can maintain the stable performances on three image databases. Therefore, the representation coefficient for each image is very sparse in new concept space. It is consistent with our understanding that each image can be presented by a linear combination only a few concepts.

5. Conclusion

In this paper, a novel semi-supervised method, called CSCC, is proposed for image representation, which has many advantages over traditional sparse coding techniques. CSCC explores the geometric structure information among the original data, and simultaneously takes advantage of the limited label information in a parameter-free manner. *Subsequently*, the kernel extension of CSCC, named KCSCC, is proposed to deal with the nonlinear distribution data. KCSCC has more discriminative power than its linear method in most cases. *Finally*, similar to SCC, our proposed algorithms only solve an eigenvalue problem and two regression problems. In comparison with traditional sparse coding algorithms, CSCC and KCSCC are also very efficient. Experimental results demonstrate that our algorithms can provide a better representation than state-of-the-art matrix factorization algorithms.

However, there are still several drawbacks existed in CSCC and KCSCC to be considered in the future work. First, our proposed algorithms cannot provide a mechanism for noise removing, thus they are not robust methods for image representation. Therefore, we can replace the ℓ-2 norm by the ℓ-2,1 norm for noisy data in future work. Second, the number of basis vectors and the cardinality play a very important role in improving the performances of our proposed algorithms. How to effectively set them is an interesting topic.

References

[1] W. Zhao, R. Chellappa, P. J. Phillips, et al, "Face recognition: A literature survey," *ACM Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399-458, 2003. Article (CrossRef Link)

[2] M. Turk, A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991. Article (CrossRef Link)

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.19, no.7, pp.711-720, 1997. Article (CrossRef Link)

[4] J. M. Lee, C. K. Yoo, S. W. Choi, et al., "Nonlinear process monitoring using kernel principal component analysis," *Chemical Engineering Science*, vol. 59, no. 1, pp. 223-234, 2004. Article (CrossRef Link)

[5] S. Mika, G. Ratsch, J. Weston, et al., "Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 623-628, 2003. Article (CrossRef Link).

[6] S. Yan, D. Xu, and B. Zhang, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40-51, 2007. Article (CrossRef Link).

[7] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319-2323, 2000. Article (CrossRef Link)

[8] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000. Article (CrossRef Link)

[9] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput*., vol. 15, no. 6, pp. 1373-1396, 2003. Article (CrossRef Link)

[10] X. He, S. Yan, Y. Hu, *et al*, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328-340, Mar. 2005. Article (CrossRef Link)

[11] X. He, M. Ji, H. Bao, "Graph embedding with constraints," in *Proc. of t the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, Pasadena, USA, vol. 9, pp. 1065-1070, 2009. Article (CrossRef Link)

[12] D. Lee, H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp.788–791, 1999. Article (CrossRef Link)

[13] J. Qian, J. Yang et.al, "Discriminative Histograms of Local Dominant Orientation (D-HLDO) for Biometric Image Feature Extraction," *Pattern Recognition*, vol. 46, no. 10, pp: 2724-2739, 2013. Article (CrossRef Link)

[14] W. Xu and Y. Gong, "Document Clustering by Concept Factorization," in *Proc. of ACM SIGIR '04*, pp: 202-209, 2004. Article (CrossRef Link)

[15] J. Qian, J. Yang, Y. Xu, "Local Structure-based Image Decomposition for Feature Extraction with Applications to Face Recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp: 3591-3603, 2013. Article (CrossRef Link)

[16] Y Chen, J. Zhang, D. Cai, et al, "Nonnegative local coordinate factorization for image representation," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 969- 979, 2013. Article (CrossRef Link)

[17] D. Cai, X. He, J. Han, et al, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548-1560, 2011. Article (CrossRef Link)

[18] H. Liu, Z. Wu, X. Li, et al, "Constrained nonnegative matrix factorization for image representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, no. 7, pp. 1299-1311, 2012. Article (CrossRef Link)

[19] Y. He, H. Lu, L. Huang, et al, "Pairwise constrained concept factorization for data representation," *Neural Networks*, vol. 52, pp.1-17, 2014. Article (CrossRef Link)

[20] Z. Li, J. Liu, H. Lu, "Structure preserving non-negative matrix factorization for dimensionality reduction," *Computer Vision and Image Understanding*, vol.117, no. 9, pp.1175-1189, 2013. Article (CrossRef Link)
[21] J. Yang, S. Yang, Y. Fu, et al, "Nonnegative graph embedding," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-8, Alaska, USA, 2008. Article (CrossRef Link)
[22] J. Wang, H. Bensmail, X. Gao, "Multiple graph regularized nonnegative matrix factorization," *Pattern Recognition*, vol. 46, no.10, pp. 2840-2847, 2013. Article (CrossRef Link)
[23] J. Wright, A. Yang, S. Sastry, et al., "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.31, no. 2, pp. 210-227, 2009. Article (CrossRef Link)
[24] Y. Qian, S. Jia, J. Zhou, et al, "Hyperspectral unmixing via L-1/2 sparsity-constrained nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp: 4282-4297, 2011. Article (CrossRef Link)
[25] M. Li, J. Tang, C. Zhao. "Active Learning on Sparse Graph for Image Annotation," *KSII Transactions on Internet & Information Systems*, vol. 6, no. 10, pp. 2650-2662, 2012. Article (CrossRef Link)
[26] J. Tang, R. Hong, S. Yan, et al. "Image annotation by k-NN sparse graph-based label propagation over noisily tagged web images," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 2, pp: 1-15, 2011. Article (CrossRef Link)
[27] H. Zheng , Q. Ye, Z. Jin. "A Novel Multiple Kernel Sparse Representation based Classification for Face Recognition," *KSII Transactions on Internet & Information Systems*, vol.8, no. 4, pp: 1483-1480, 2014. Article (CrossRef Link).
[28] D. Cai, X. Wang, and X. He, "Probabilistic dyadic data analysis with local and global consistency," in *Proc. of the 26th Annual International Conference on Machine Learning*, pp. 105-112, 2009. Article (CrossRef Link)
[29] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no.6, pp. 902 - 913 2011. Article (CrossRef Link)
[30] J. Wang, J. Yang, K Yu, et al, "Locality-constrained linear coding for image classification," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* pp:3360-3367, San Francisco, 2010. Article (CrossRef Link)
[31] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 2272-2279, Kyoto, 2009. Article (CrossRef Link)
[32] S Gao, I W H Tsang, L T Chia, "Laplacian sparse coding, hypergraph laplacian sparse coding, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 92-104, 2013. Article (CrossRef Link)
[33] Zheng M, Bu J, Chen C, et al, "Graph regularized sparse coding for image representation," *IEEE Transactions on Image Processing*, vol. 20,no. 5, pp. 1327-1336, 2011. Article (CrossRef Link)
[34] D. Cai, H. Bao, X. He, "Sparse concept coding for visual analysis," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2905-2910, Colorado, USA, 2011. Article (CrossRef Link)
[35] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, *et al*, "Learning with local and global consistency," in *Proc. of Advances in Neural Information Processing Systems*, pp. 321-328, Vancouver, Canada, 2004. Article (CrossRef Link)
[36] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proc. of the twentieth Internation Conference on Machine Learning*, pp. 912-919, Washington, DC, United states, 2003. Article (CrossRef Link)
[37] Donoho D. L., "For most large underdetermined systems of linear equations the minimal L1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797-829, 2006. Article (CrossRef Link)

[38] Candes E J, Romberg J K, Tao T, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207-1223, 2006. Article (CrossRef Link)
[39] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407-499, 2004. Article (CrossRef Link)
[40] C. C. Paige and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43-71, 1982. Article (CrossRef Link)
[41] J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83-85, 2001. Article (CrossRef Link)

Zhenqiu Shu received the B.Sc. degree from University of South China, Hengyang, China, in 2008 and the M.S. degree Kunming University of Science and Technology, Kunming, China, in 2011. From 2011 to now, he is working toward his Ph.D. degree in Pattern Recognition and Intelligent Systems from Nanjing University of Science and Technology (NUST), Jiangsu, China. His research interests include machine learning, data mining, and pattern recognition.

Chunxia Zhao received the B.E., M.S. and Ph.D. degrees from Harbin Institute of Technology, Harbin, China, in 1985, 1988, 1998, respectively, both in the Department of Electrical Engineering and Computer. She is a professor in the Department of Computer Science, Nanjing University of Science and Technology. Her current interests are in the areas of robots, computer vision, and pattern recognition.

Pu Huang received his B.S. and M.S. degrees in computer applications from Yangzhou University, PR China, in 2007 and 2010, respectively. He is currently pursuing the Ph.D. degree in Pattern Recognition and Intelligent Systems at Nanjing University of Science and Technology (NUST), China. His research interests include pattern recognition, computer vision and machine learning.