A Locality-Constrained and Label Embedding Dictionary Learning Algorithm for Image Classification

Zhengming Li, Zhihui Lai, Yong Xu, *Senior Member, IEEE*, Jian Yang, and David Zhang, *Fellow, IEEE*

***Abstract*— Locality and label information of training samples play an important role in image classification. However, previous dictionary learning algorithms do not take the locality and label information of atoms into account together in the learning process, and thus their performance is limited. In this paper, a discriminative dictionary learning algorithm, called the locality-constrained and label embedding dictionary learning (LCLE-DL) algorithm, was proposed for image classification. First, the locality information was preserved using the graph Laplacian matrix of the learned dictionary instead of the conventional one derived from the training samples. Then, the label embedding term was constructed using the label information of atoms instead of the classification error term, which contained discriminating information of the learned dictionary. The optimal coding coefficients derived by the locality-based and label-based reconstruction were effective for image classification. Experimental results demonstrated that the LCLE-DL algorithm can achieve better performance than some state-of-the-art algorithms.**

***Index Terms*— Dictionary learning, label embedding, locality constrained, profile, sparse coding.**

I. Introduction

In THE past several years, dictionary learning for sparse coding has been widely used in image classification. As a fundamental problem in sparse coding, dictionary learning has attracted a lot of attention in recent years. Some recent reviews on the sparse coding and dictionary learning can be found in [1]–[3]. In general, the main goal of dictionary learning is to learn the atoms from training samples. The test samples can be sparsely represented by the learned atoms, by which the classification task can be performed based on the reconstruction error and/or the sparse coefficients.

Many previous studies [4]–[6] used the original training samples as a dictionary to reconstruct the test samples, and achieved impressive results in comparison with many well-known face recognition algorithms. Recently, many works demonstrated that learning a dictionary from the training samples can lead to better performance in many image-related applications, such as image restoration and classification. The K-SVD algorithm is one of the well-known dictionary learning algorithms [7]. Actually, it is a generalized k-means clustering algorithm [8]. However, the K-SVD algorithm is not suitable for classification tasks, because it only requires that the learned dictionary should well reconstruct the training samples. In order to improve the discriminative ability of the learned dictionary, Pham and Venkatesh [9] used the results of a linear classifier to iteratively update the K-SVD algorithm. Thus, a dictionary containing powerful discriminative and reconstructive ability can be obtained for image classification. In order to further enhance the discriminative ability of the learned dictionary, a discriminative K-SVD (D-KSVD) algorithm was proposed by Zhang and Li [10]. Although those dictionary learning algorithms achieve excellent performance for image classification, they, generally, do not ensure locality preservation and, thus, are not optimal, since the data may lie on the nonlinear manifold embedded in a very high-dimensional ambient space [11], [12], and thus the classification performance will be degraded.

Locality information of data has been observed to be a key issue in many real applications, especially in sparse coding and dictionary learning. Locality is more essential than sparsity, since locality leads to sparsity but not necessary vice versa [13]. Therefore, more and more researchers focused on locality preservation in sparse coding and dictionary learning. The basic idea of learning a dictionary is to encode the training samples while incorporating some locality constraints, which ensures that similar training samples tend to have similar coding coefficients. According to the manifold assumption, Yu *et al.* [13] proposed a local coordinate coding (LCC) algorithm by taking advantage of the local geometric structure of the training samples. Although the LCC algorithm achieves

Manuscript received November 1, 2014; revised November 18, 2015; accepted December 6, 2015. Date of publication December 24, 2015; date of current version January 17, 2017. This work was supported in part by the Natural Science Foundation of China under Grant 61370163, Grant 61375012, and Grant 61573248, and in part by the Shenzhen Council for Scientific and Technological Innovation under Grant JCYJ20140904154630436 and Grant JCYJ20150324141711637. *(Corresponding author: Yong Xu.) (Zhengming Li and Zhihui Lai contributed equally to this work.)*

Z. Li is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China, and also with the Industrial Training Center, Guangdong Polytechnic Normal University, Guangzhou 510665, China (e-mail: lizhengming2004@126.com).

Z. Lai is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: lai_zhi_hui@163.com).

Y. Xu is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: laterfall@hitsz.edu.cn).

J. Yang is with the School of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: csjyang@mail.njust.edu.cn).

D. Zhang is with the Biometrics Research Centre, Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csdzhang@comp.polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2015.2508025

the state-of-the-art performance with a linear classifier for image classification, the main drawback is the high computational cost and it is not suitable for large-scale learning problems. In order to reduce the computational complexity, Wang *et al.* [14] proposed a locality-constrained linear coding (LLC) algorithm by using the distances between the bases (atoms) and the training samples to select the k-nearest neighbor atoms for coding, and set the coding coefficients of other bases to zero. Recently, several variants of the LLC algorithm have been proposed by adding some constraints or specific applications, such as the specific class LLC algorithm [15], multilevel LLC algorithm [16], nonnegative-constrained LLC algorithm [17], and low-rank-constrained LLC algorithm [18].

Moreover, Zhou and Barner [19] proposed a locality-constrained dictionary learning algorithm by selecting a few latent landmark points (a subset of the training samples), and used them to select the nearest neighbor bases for large-scale learning problems. However, the selection of the nearest bases is sensitive to noise in the training samples. In other words, two training samples of the same class may select different nearest bases if one of them is contaminated by noise. Thus, there is a conflict when the training samples of the same class have similar coding coefficients in the ideal case. On the other hand, Gao *et al.* [20] and Zheng *et al.* [21] proposed the Laplacian sparse coding algorithm, which can exploit the dependence among the local features. Based on the assumption that data points are distributed on the same manifold, Ramamurthy *et al.* [22] proposed a manifold projection method to improve the efficiency of sparse coding. In order to identify the representation of low-dimensional subspaces from the high-dimensional and nonnegative data, Jing *et al.* [23] proposed a dictionary learning algorithm based on nonnegativity and sparsity constraints together for spectral clustering. However, since the l_1 regularization term is added to the objective functions of those algorithms, it can lead to high computational cost and it is not suitable for large-scale training samples. Furthermore, those algorithms do not use the label information of the training samples, and thus the classification performance will also be degraded.

In order to improve the performance of the learned dictionary, many studies have focused on the properties of atoms. The dictionary learning algorithms by using the coherence of atoms were proposed in [24]–[27]. Those algorithms demonstrate that the smaller the coherence of atoms, the better the reconstruction ability of the learned dictionary. Since the locality information is not considered in the dictionary learning process, there is a limitation in improving the discriminative ability of the learned dictionary. Recently, Shaban *et al.* [28] exploited the local similarities among atoms and used them to measure the global similarities of the training samples. As a result, the influence of the noise can be reduced to some extent. In addition, Jiang *et al.* [8] proposed a label consistent K-SVD (LC-KSVD2) dictionary learning algorithm, which associated the label information with the atoms to improve the classification performance. Unfortunately, in the dictionary learning process, those algorithms have not taken the locality and label information of atoms into account together. Thus, the coding coefficients of the training samples may greatly vary. It also potentially degrades the classification performance. Recently, Lu *et al.* [29] used the row vector of the coding coefficient matrix to determine which atoms are used in dictionary learning. Sadeghi *et al.* [30] defined the row vector of the coding coefficient matrix as the profile. It indicated which training samples used the corresponding atoms to encode. This inspired us to design a new algorithm to improve the discriminative ability of the learned dictionary.

In this paper, we first assigned the label information to each atom and constructed the label embedding of atoms by using the labels of atoms and profile matrix. It encouraged the same class atoms to have similar profiles, and it also forced the coding coefficient matrix to be block-diagonal. Moreover, we calculated the local similarities among the atoms to reflect the local geometric properties of the learned dictionary. Then, the locality constraint of atoms was constructed using the automatic learning strategy to inherit the manifold structure of the training samples. In addition, we used the double reconstruction terms for dictionary learning, and then the locality reconstruction and label reconstruction were fitted at the same time. Therefore, it was expected to learn a discriminative dictionary for image classification.

Our main contributions are as follows.

First, we constructed the label embedding of atoms to encourage the same class atoms to have similar profiles. Moreover, it also can force the coding coefficient matrix to be block-diagonal. Thus, we can obtain the optimal representation of the training samples.

Second, we constructed the locality constraint of atoms to ensure that similar profiles encouraged the corresponding atoms to be similar. In addition, the learned dictionary was adaptive to the training samples by the derived-graph Laplacian matrix, which, in turn, inherited the manifold structure of the training samples.

Third, we utilized the double reconstruction terms in the locality-constrained and label embedding dictionary learning (LCLE-DL) algorithm to ensure that the locality-based and label-based coding coefficients are as approximate to each other as possible. Experimental results on the five image databases show that the learned dictionary preserved good reconstruction ability and discriminating ability.

The rest of this paper is organized as follows. Section II introduces the LCLE-DL algorithm. Section III presents the optimization methods of the LCLE-DL algorithm. Section IV gives the relationships between the LCLE-DL algorithm and some previous algorithms. Section V presents the experimental results and analyses. Section VI presents the conclusions.

II. LCLE-DL Algorithm

Suppose we are given a set of N training samples in an n dimension $Y = [Y_1, \ldots, Y_C] = [y_1, y_2, \ldots, y_N] \in \Re^{n \times N}$, C is the class number of the training samples. Y_i is a matrix composed of all training samples of the ith class. Thus, the label matrix of training samples Y can be defined as $H = [h_1, \ldots, h_N] \in \Re^{C \times N} (h_i = [0, \ldots, 1, \ldots, 0]^T \in \Re^C$, and only the jth entry of h_i is nonzero, which indicates that training sample y_i comes from the jth class).

$$
\begin{aligned}
y_1 &= d_1x_{1,1} + d_2x_{2,1} + \cdots + d_ix_{i,1} \cdots + d_Kx_{K,1} \\
y_2 &= d_1x_{1,2} + d_2x_{2,2} + \cdots + d_ix_{i,2} \cdots + d_Kx_{K,2} \\
&\cdots\cdots \\
y_N &= d_1x_{1,N} + d_2x_{2,N} + \cdots + d_ix_{i,N} \cdots + d_Kx_{K,N}
\end{aligned}
$$

Fig. 1. Basic model of dictionary learning.

$D = [d_1, \ldots, d_K] \in \Re^{n\times K}$ is the learned dictionary from training samples Y. K is the number of atoms. In our algorithm, learned dictionary D has the same categories as the training samples, and we also assume that each class has the same atom number f. Thus, the number of atoms K is the integral multiple of C, that is, $K = f \times C$. $X = [x_1, x_2, \ldots, x_N] \in \Re^{K\times N}$ is the coding coefficient matrix. $x_i = [x_{1i}, x_{2i}, \ldots, x_{Ki}]^T (i = 1, \ldots, N)$ is the coding vector of the training sample y_i corresponding to dictionary D. Following [29], the basic model of dictionary learning can be rewritten as in Fig. 1.

According to [30], the ith row vector of coding coefficient matrix X is called the profile of atom d_i. Then, we can define vector $\hat{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iN}]^T (i = 1, \ldots, K)$ as the profile of atom d_i. The red rectangle in Fig. 1 shows profile $\hat{x}_i$, and it indicates which training samples use atom d_i to encode. Thus, the profile matrix can be defined as $X^T = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_K] \in \Re^{N\times K}$, and it is a transposed matrix of coding coefficient matrix X.

According to the definition of the profile, the basic model of dictionary learning can be rewritten as follows:

$$Y = d_1(\hat{x}_1)^T + \cdots d_i(\hat{x}_i)^T + \cdots d_j(\hat{x}_j)^T + \cdots + d_K(\hat{x}_K)^T. \tag{1}$$

It can be seen that there are one-to-one correspondences between the profiles and the atoms. If profile $\hat{x}_i$ is a zero vector, atom d_i has no contribution to reconstruct training samples Y. Otherwise, it has some contributions to reconstruct training samples Y. Thus, the profiles can be used to measure the contributions of atoms to reconstruct the training samples. Moreover, the similar profiles can encourage the corresponding atoms to be similar, and the same class atoms also tend to have similar profiles. Therefore, we can use the profile matrix and properties of atoms (e.g., label and locality) to construct the discriminative constraint term.

A. Label Embedding of Atoms

As presented in [8], if we assign a label to an atom and construct a label constraint term, it can improve the discriminative ability of the learned dictionary. In particular, for the classification task, the learned dictionary should have powerful discriminative ability for different classes. That is, in ideal conditions, some atoms should only reconstruct the training samples of the same class [31]. Therefore, it is reasonable to assume that these atoms belong to the same class. Inspired by the specific class dictionary learning algorithms, we can assign the labels of the training samples to atoms. How to assign a label to an atom is presented as follows.

1) We used a dictionary learning algorithm (e.g., K-SVD) to learn subdictionary D_i by using the ith class training samples. If atom $d_i \in D_i$, the label vector of atom d_i can be defined as $b_i = [0, \ldots 1, \ldots, 0] \in \Re^C$. The sole nonzero value of b_i occurs at its ith element, which indicates that atom d_i belongs to the ith class.
2) For each class, we learned subdictionary $D_i (i = 1, 2, \ldots, C)$. Then, the learned dictionary can be denoted by $D = [D_1, \ldots, D_C]$. It contains all the atoms of C classes, and D_i is the subdictionary of the ith class. Thus, the label matrix B of dictionary D can be defined as $B = [b_1, \ldots, b_K]^T \in \Re^{K\times C}$.

In [32], in order to group data $\{y_i\}_{i=1}^N$ into C clusters $\{F_j\}_{j=1}^C$, the authors defined a cluster indicator matrix $S \in^{N\times C}$ as follows:

$$S = \{s_{i,j}\}_{N\times C}, \quad \text{if } y_i \in F_j,\ s_{i,j} = 1, \quad \text{else } s_{i,j} = 0. \tag{2}$$

Then, a weighted cluster indicator matrix Q was defined as follows:

$$Q = [Q_1, \ldots, Q_C] = S(S^TS)^{-\frac{1}{2}}. \tag{3}$$

Following [32], we used label matrix $B \in \Re^{K\times C}$ of dictionary D to construct a weighted label matrix G as follows:

$$G = B(B^TB)^{-\frac{1}{2}} \in \Re^{K\times C}. \tag{4}$$

In order to encourage the same class atoms to have similar profiles, we tried to use the profile matrix and labels of atoms to construct a label embedding term. Then, we can define the label embedding of atoms as follows:

$$\min_V \mathrm{Tr}(V^TGG^TV) = \min_V \mathrm{Tr}(V^TUV) \tag{5}$$

where $U = GG^T \in \Re^{K\times K}$ is the scaled label matrix of dictionary D. Equation (5) ensures that the same class atoms have similar profiles. The proof is presented in Appendix A. Furthermore, it also forces the coding coefficient matrix to be block-diagonal. Thus, the coding coefficient matrix can be regarded as the optimal representation of the training samples [33].

Because scaled label matrix U is constructed according to the labels of the training samples, it is a constant in our algorithm. According to (4) and the definition of the weighted label matrix, we can infer that scaled label matrix U has a block-diagonal structure. Therefore, minimizing the label embedding term of atoms can force coding coefficient matrix V to be block-diagonal. Fig. 2 shows that coding coefficient matrix V on the PIE face database is nearly block-diagonal. According to [31], in ideal conditions, some atoms should only reconstruct the training samples of the same class. Therefore, it is reasonable to assume that the learned dictionary should contain different class atoms, and each class atoms should reconstruct one class of the training samples. Thus, the label embedding of atoms can also encourage the training samples of the same class to have more similar coding coefficients than other classes.

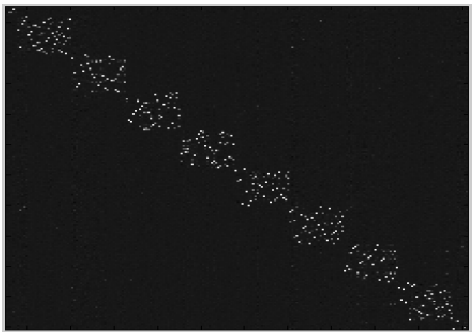

Fig. 2. Illustration of coding coefficient matrix V learned using the proposed algorithm on the PIE database.

B. Locality Constraint of Atoms

Locality information plays an important role in the dictionary learning and sparse coding algorithms. Because the training samples often include noise and outliers in practical applications, they may not calculate the true locality information of the training samples. According to [28] and [34], if a dictionary is learned by using the k-means algorithm, the atoms can trace the manifold structure of the training samples. Therefore, we tried to use the locality information of the atoms to inherit the structure information of the training samples. Since the atoms are learned from the training samples by using a holistic algorithm, it is more robust to noise and outliers than the training samples. In other words, the learned dictionary is usually more stable than a single original training sample, and it is able to smooth and reduce the influence of the noise of original training samples and outliers. Moreover, the atoms can be updated with the dictionary learning processing. Therefore, the locality information of the atoms can somewhat overcome the problem caused by the noise and the outliers. Following [28], we constructed a nearest neighbor graph M of dictionary D as

$$M_{i,j} = \begin{cases} \exp\left(-\dfrac{\|d_i - d_j\|_2}{\delta}\right) & \text{if } d_j \in k\text{NN}(d_i) \\ 0 & \text{else} \end{cases} \tag{6}$$

where δ is a parameter, $k\text{NN}(d_i)$ denotes the k-nearest neighbors of atom d_i, and $M_{i,j}$ reflects the similarity between atoms d_i and d_j. If atom d_i is connected with atom d_j, we can assume that the two atoms are close. In order to better represent the locality information of atoms, we define a graph Laplacian matrix L by using nearest neighbor graph M as follows:

$$L = T - M, T = \text{diag}(t_1, \ldots, t_K), \quad \text{and} \quad t_i = \sum_{j=1}^{K} M_{i,j}. \tag{7}$$

Since the K-SVD algorithm can be regarded as a generalized k-means clustering algorithm, we used it to initialize a subdictionary for each class. Then, the initialization dictionary D can be used to construct a graph Laplacian matrix L, and it can be updated with dictionary D. Thus, the graph Laplacian matrix L can be used to inherit the manifold structure of the training samples. Moreover, it is more robust to noise and outliers than directing using the training samples. Since the profiles and atoms have a one-to-one correspondence, we used the profile matrix X^T and the graph Laplacian matrix L to preserve the locality information of the learned dictionary. Following [21], a reasonable criterion for choosing a good graph Laplacian matrix L is to minimize the following equation:

$$\frac{1}{2}\sum_{i=1}^{K}\sum_{j}^{K}(\hat{x}_i - \hat{x}_j)^2 M_{i,j} = \text{Tr}(X^T L X). \tag{8}$$

Equation (8) indicates that the similar profiles encourage the corresponding atoms to be similar. The proof is presented in Appendix B.

C. Objective Function of the LCLE-DL Algorithm

In order to obtain a discriminative dictionary, we combined the label embedding and locality constraint of atoms to learn a dictionary. Then, we used a regularization term to ensure that the transformation between the label embedding and locality constraint is mutual. Furthermore, it also can force the locality-based and label-based coding coefficients to be as close as possible. Thus, we defined the objective function of the LCLE-DL algorithm as follows:

$$\begin{aligned} \min_{D,X,V,L} & \|Y - DX\|_2^2 + \alpha \text{Tr}(X^T L X) + \|Y - DV\|_2^2 \\ & + \beta \text{Tr}(V^T U V) + \gamma \|X - V\|_2^2 \\ \text{s.t. } & \|d_i\|^2 = 1, \quad i = 1, \ldots, K \end{aligned} \tag{9}$$

where $X \in \Re^{K \times N}$ and $V \in \Re^{K \times N}$ are the coding coefficient matrices, $\|Y - DX\|_2^2$ and $\|Y - DV\|_2^2$ denote the reconstruction error terms, and $\|X - V\|_2^2$ is a regularization term used to transfer the label constraint to/from the locality constraint. α, β, and γ are the regularization parameters. The first and second terms encode the reconstruction under the locality constraint. Moreover, the second term represents the locality information of dictionary D, which can inherit the manifold structure of the training samples. The third and fourth terms encode the reconstruction under label embedding. The fourth term represents label embedding V as the profiles of dictionary D, which forces the atoms in dictionary D from the same class to have very similar profiles. The fifth term is a regularization of two coding coefficients, and it ensures that the transformation between the label embedding and locality constraint is mutual.

Since scaled label matrix U is directly constructed using the label matrix of the learned dictionary, scaled label matrix U has the block-diagonal structure with strong discriminative information. Scaled label matrix U can be viewed as a supervised graph Laplacian matrix, so $\min \text{Tr}(V^T U V)$ means that coding coefficient matrix V is the spectral embedding of the graph Laplacian matrix with strong discriminative information. At the same time, $\min \|X - V\|_2^2$ indicates that coding coefficient matrix X should approximate to coding coefficient matrix V. Thus, the discriminative information contained in coding coefficient matrix V is transformed to coding coefficient matrix X. Since coding coefficient matrix X is also the spectral embedding of graph Laplacian matrix L, $\min \text{Tr}(X^T L X)$ forces graph Laplacian matrix L to be the

graph Laplacian matrix of embedding matrix V. Note that graph Laplacian matrix L is constructed by the learned dictionary's similarity; therefore, $\min \mathrm{Tr}(X^T LX)$ also forces graph Laplacian matrix L to be with the similar discriminative structure as scaled label matrix U, since the model requires $X \rightarrow V$. Therefore, the discriminative information can be transformed to coding coefficient matrix X and graph Laplacian matrix L for learning a dictionary with strong discriminative ability. Thus, $\min \|Y - DX\|_2^2$ will lead to a dictionary with strong discriminative ability, since both the dictionary's structure/similarity and the representation coefficient are taken into account. The proposed model can not only sufficiently use the discriminative information and transform it to the learned dictionary but also endure a certain error, which will further enhance the learned dictionary generality in representation. There are three specific features that distinguish the LCLE-DL algorithm from some previous dictionary learning algorithms.

1) Instead of using the locality information of the training samples, the locality information of the learned dictionary is used to inherit the intrinsic geometric properties of the training samples to some extent. It is more robust to noise than directly using the locality information of the training samples.
2) Instead of directly constructing the classification error term by using the label information of training samples, we constructed the label embedding of atoms to force the coding coefficient matrix to be block-diagonal.
3) Differing from the K-SVD algorithm and its variations, the dual reconstruction terms of the LCLE-DL algorithm ensure that the locality-based and label-based coding coefficients are as close as possible, which can transfer the label information and locality information of atoms to the coding coefficients.

III. Optimization of the Objective Function

In general, the atoms of the learned dictionary can be obtained one by one by using the K-SVD algorithm. The objective function of the LCLE-DL algorithm can be solved by a closed form solution when some variables are fixed. Therefore, an iteration algorithm was used to solve our objective function, and the computational complexity was decreased. Since there were four variables in our objective function, we first used the K-SVD algorithm to initialize specific class dictionary D and the corresponding coding coefficient matrix X. Thus, we constructed graph Laplacian matrix L and scaled label matrix U. Then, we fixed dictionary D and coding coefficient matrix X, and coding coefficient matrix V was obtained. Next, we fixed dictionary D, graph Laplacian matrix L, and coding coefficient matrix V, and coding coefficient matrix X was obtained. Finally, we fixed coding coefficient matrices X and V, and dictionary D was obtained, and then graph Laplacian matrix L was constructed using dictionary D.

A. Initialization of the LCLE-DL Algorithm

In order to obtain scaled label matrix U and initialization graph Laplacian matrix L, we used the K-SVD algorithm to learn subdictionary D_i and coding matrix X_i for the ith class training samples Y_i. Thus, we obtained initialization dictionary $D = [D_1, D_2, \ldots, D_C]$ and coding coefficient matrix $X = [X_1, X_2, \ldots, X_C]$. Then, we constructed the label matrix B of dictionary D according to the label matrix of training samples. Next, we used (4) to calculate the weighted label matrix G of dictionary D, and scaled label matrix U was obtained using $U = GG^T$. Moreover, the initialization graph Laplacian matrix L was constructed using (6) and (7).

B. Learning the Coding Coefficient Matrices V and X

In order to obtain the coding coefficient matrix V, we ignored the constant terms with respect to V in (9). Thus, the objective function of the LCLE-DL algorithm became

$$\min_V \|Y - DV\|_2^2 + \beta \mathrm{Tr}(V^T UV) + \gamma \|X - V\|_2^2. \quad (10)$$

The optimal solution V was derived by taking the first-order derivation of (10) and setting it to zero. This led to

$$-D^T Y + D^T DV + \beta UV + \gamma V - \gamma X = 0. \quad (11)$$

Thus, the optimal V was obtained as follows:

$$V = (D^T D + \beta U + \gamma I)^{-1}(D^T Y + \gamma X). \quad (12)$$

Similarly, ignoring the constant terms with respect to X in (9), we also obtained the other optimization problem as follows:

$$\min_X \|Y - DX\|_2^2 + \alpha \mathrm{Tr}(X^T LX) + \gamma \|X - V\|_2^2. \quad (13)$$

In a similar way, the optimal X was obtained as follows:

$$X = (D^T D + \alpha L + \gamma I)^{-1}(D^T Y + \gamma V) \quad (14)$$

where I is an identity matrix.

C. Learning Dictionary D and Graph Laplacian Matrix L

In order to learn an optimal dictionary D, we supposed that other variables were given in (9). Then, the objective function of the LCLE-DL algorithm became

$$\min_D \|Y - DX\|_2^2 + \|Y - DV\|_2^2 \quad \text{s.t. } \|d_i\|^2 = 1, \quad i = 1, \ldots, K. \quad (15)$$

Thus, it became a least square problem with quadratic constraints and could be solved by many methods. As suggested in [21], the Lagrange dual function was used to solve (15), and then we had

$$g(\lambda) = \inf\left(\|Y - DX\|_2^2 + \|Y - DV\|_2^2 + \sum_{i=1}^{K} \lambda_i (\|d_i\|^2 - 1)\right) \quad (16)$$

where $\lambda = [\lambda_1, \ldots, \lambda_K], i \in [1, \ldots, K]$, and λ_i is the Lagrange multiplier of the ith equality constraint $(\|d_i\|^2 - 1 = 0)$. We defined a diagonal matrix $\Lambda \in \Re^{K \times K}$, and diagonal entry $\Lambda_{ii} = \lambda_i (i = 1, \ldots, K)$ for all i. Equation (16) was derived as follows:

$$L(D, \lambda) = \|Y - DX\|_2^2 + \|Y - DV\|_2^2 + \mathrm{Tr}(D^T D\Lambda) - \mathrm{Tr}(\Lambda). \quad (17)$$

TABLE I
LCLE-DL ALGORITHM

Input: Training samples $Y=[y_1,y_2,\cdots,y_N]\in\Re^{m\times N}$, label matrix H, parameters: α, β, γ, k, and δ, iterations $T_{\max}$.
Output: Dictionary D, coding coefficient matrixes X and V

Step 1: For $i=1:C$
Learning sub-dictionary D_i and coding coefficient matrix X_i by using the K-SVD algorithm on subset Y_i.
End
Step2: Obtaining initialization dictionary $D^0=[D_1,D_2,\cdots,D_C]$ and initialization coding coefficient matrix $X^0=[X_1,X_2,\cdots,X_C]$.
Step 3: Constructing label matrix B of dictionary D^0 according to label matrix H.
Step 4: Calculating weighted label matrix G, and then calculating scaled label matrix U by using $U=GG^T$.
Step 5: For $i=1:T_{\max}$
Calculating graph Laplacian matrix L_i by using dictionary D^{i-1}.
Calculating coding coefficient matrix V^i by using $\left(\left(D^{i-1}\right)^T D^{i-1}+\beta U+\gamma I\right)^{-1}\left(\left(D^{i-1}\right)^T Y+\gamma X^{i-1}\right)$.
Calculating coding coefficient matrix X^i by using $\left(\left(D^i\right)^T D^i+\alpha L^i+\gamma I\right)^{-1}\left(\left(D^i\right)^T Y+\gamma V^i\right)$.
Calculating dictionary D^i by using $Y\left(\left(X^{i-1}\right)^T+\left(V^i\right)^T\right)\left(X^{i-1}\left(X^{i-1}\right)^T+V^i\left(V^i\right)^T\right)^{-1}$
End
Step 6: Obtaining $D=D^{T_{\max}}$, $X=X^{T_{\max}}$, and $V=V^{T_{\max}}$

In order to obtain the optimal solution of dictionary D, we took the first-order partial derivatives of (17) and set it to zero. Then, the optimal dictionary was obtained

$$D^*=Y(X^T+V^T)(XX^T+VV^T+\Lambda)^{-1}. \tag{18}$$

For the sake of reducing the computational complexity, the optimal dictionary D was derived using (19) by discarding Λ. The optimization of which is shown in [21]

$$D=Y(X^T+V^T)(XX^T+VV^T)^{-1}. \tag{19}$$

Then, we used (6) and (7) to construct the graph Laplacian matrix L. The proposed LCLE-DL algorithm is shown in Table I.

D. Classification Method

Although coding coefficient matrix X and coding coefficient matrix V are different in our algorithm, we used the regularization term to force them to be as close as possible. Therefore, we only used the coding coefficient matrix X to perform the task of image classification in the LCLE-DL algorithm. Following [9], a linear classifier method was used for image classification in the LCLE-DL algorithm.

First, a classifier parameter W_x was obtained using the coding coefficient matrix X and label matrix H of the training samples as follows:

$$W_x=HX^T(XX^T+I)^{-1}. \tag{20}$$

Second, for a test image $\tilde{y}_i$, its sparse representation coefficient vector $\tilde{x}_i$ with the learned dictionary D was obtained using the Orthogonal Matching Pursuit algorithm [35] algorithm. Then, a label vector l_x was calculated using $W_x\tilde{x}_i$. Finally, the label of test sample $\tilde{y}_i$ was the index corresponding to the largest element of the label vector l_x.

IV. Relationships Between Our Algorithm and Some Previous Works

In this section, we established the relationships between the LCLE-DL algorithm and some previous dictionary learning and sparse coding algorithms, such as the D-KSVD, Graph regularized Sparse Codes algorithm (Graph-SC), and LLC algorithms.

A. Connection to the D-KSVD Algorithm

The D-KSVD algorithm can be regarded as an extension of the K-SVD algorithm. It adds the classification error term to the objective function of the K-SVD algorithm. Moreover, the LC-KSVD2 algorithm can also be regarded as an extension of the D-KSVD algorithm. It adds the discriminative sparse code error term to the objective function of the D-KSVD algorithm.

In [10], the objective function of the D-KSVD algorithm is

$$\begin{aligned}&\min_{D,W,V}\|Y-DV\|_2^2+\alpha\|H-WV\|_2^2\\&\text{s.t. }\|V\|_0\le\xi\end{aligned} \tag{21}$$

where $\|H-WV\|_2^2$ is the classification error term, $H\in\Re^{C\times N}$ is the label matrix of training samples Y, $W\in\Re^{C\times K}$ is the classifier parameter, $V\in\Re^{K\times N}$ is the coding coefficient matrix, and ξ is the sparsity factor.

According to (21), if matrix WV is equal to matrix H, the second term of (21) can achieve the minimum value. In ideal conditions, the same class training samples should be well reconstructed by the same class atoms [31], [33]. Therefore, we replaced matrix W with weighted label matrix G^T, and the second term of (21) can achieve a minimum value. Since the number of atoms $K=f\times C$, matrix H^TW was rewritten as

$$H^TW=H^TG^T=\begin{bmatrix}Z_1&0&\cdots&0\\0&Z_2&\cdots&0\\\vdots&\vdots&\cdots&\vdots\\0&0&\cdots&Z_C\end{bmatrix}\in\Re^{N\times K} \tag{22}$$

where $Z_1=Z_2=\cdots Z_C=(f)^{-0.5}Z\in\Re^{(N/C)\times f}$, and all the elements of matrix Z were set to 1. Moreover, the block matrix of coding coefficient matrix V was rewritten as

$$V=\begin{bmatrix}P_{1,1}&P_{1,2}&\cdots&P_{1,C}\\P_{2,1}&P_{2,2}&\cdots&P_{2,C}\\\vdots&\vdots&\vdots&\vdots\\P_{C,1}&P_{C,2}&\cdots&P_{C,C}\end{bmatrix}\in\Re^{K\times N} \tag{23}$$

where

$$P_{ij} = \begin{bmatrix} v_{(i-1)\times f+1,(j-1)\times\frac{N}{C}+1} & \cdots & v_{(i-1)\times f+1,(j-1)\times\frac{N}{C}+\frac{N}{C}} \\ \vdots & \vdots & \vdots \\ v_{(i-1)\times f+f,(j-1)\times\frac{N}{C}+1} & \cdots & v_{(i-1)\times f+f,(j-1)\times\frac{N}{C}+\frac{N}{C}} \end{bmatrix} \tag{24}$$

where $P_{i,j} \in \Re^{f\times(N/C)}(i, j = 1, 2, \ldots, C)$. Thus, from (22)–(24), we obtained

$$H^T WV = H^T G^T V = (f)^{-0.5}\begin{bmatrix} Z\times P_{1,1} & Z\times P_{1,2} & \cdots & Z\times P_{1,C} \\ Z\times P_{2,1} & Z\times P_{2,2} & \cdots & Z\times P_{2,C} \\ \vdots & \vdots & \vdots & \vdots \\ Z\times P_{C,1} & Z\times P_{C,2} & \cdots & Z\times P_{C,C} \end{bmatrix}. \tag{25}$$

Then

$$\begin{aligned} &\mathrm{Tr}(H^T WV) \\ &\quad = \mathrm{Tr}(H^T G^T V) \\ &\quad = (f)^{-0.5}\mathrm{Tr}(Z\times P_{1,1} + Z\times P_{2,2} + \cdots + Z\times P_{C,C}). \end{aligned} \tag{26}$$

Since

$$\mathrm{Tr}(Z\times P_{i,i}) = \sum_{r=(i-1)\times f+1,}^{(i-1)\times f+f}\ \sum_{s=(i-1)\times\frac{N}{C}+1}^{(i-1)\times\frac{N}{C}+\frac{N}{C}} v_{r,s}. \tag{27}$$

Thus, we obtained

$$\begin{aligned} \mathrm{Tr}(H^T WV) &= \mathrm{Tr}(H^T G^T V) \\ &= (f)^{-0.5}\sum_{i=1}^{C}\ \sum_{r=(i-1)\times f+1,}^{(i-1)\times f+f}\ \sum_{s=(i-1)\times\frac{N}{C}+1}^{(i-1)\times\frac{N}{C}+\frac{N}{C}} v_{r,s}. \end{aligned} \tag{28}$$

It was found that (28) was just the sum of the coefficients of the block-diagonal of the coding coefficient matrix, and the size of block was $f \times (N/C)$. According to Appendix A, we inferred that the essence of the label embedding of atoms was as follows:

$$\begin{aligned} \mathrm{Tr}(V^T UV) &= \sum_{i=1}^{C}\sum_{j=1}^{N}(v_{(i-1)f+1,j} + \cdots + v_{\mathrm{if},j})^2 \\ &= \sum_{i=1}^{C}\begin{pmatrix}(v_{(i-1)f+1,1} + \cdots + v_{\mathrm{if},1})^2 \\ + \cdots + (v_{(i-1)f+1,N} + \cdots + v_{\mathrm{if},N})^2\end{pmatrix}. \end{aligned} \tag{29}$$

Therefore

$$\begin{aligned} &\mathrm{Tr}(V^T W^T WV) \\ &\quad = \mathrm{Tr}(V^T GG^T V) = \mathrm{Tr}(V^T UV) \\ &\quad = \sum_{i=1}^{C}((v_{(i-1)f+1,1} + \cdots + v_{\mathrm{if},1})^2 \\ &\qquad + \cdots + (v_{(i-1)f+1,N} + \cdots + v_{\mathrm{if},N})^2). \end{aligned} \tag{30}$$

Moreover, (31), shown at the bottom of this page, shows that the essence of the D-KSVD algorithm is to minimize the coefficients of the coding matrix. The essence of the label embedding term of the LCLE-DL algorithm is also to minimize the coefficients of the coding matrix. The main difference is that the D-KSVD algorithm reduces the sum of part coefficients, which is the block-diagonal of the coding coefficient matrix.

B. Connection to the Graph-SC Algorithm

The Graph-SC algorithm uses the locality information of the training samples in the coding and dictionary learning procedure. In [21], the objective function of the Graph-SC algorithm is

$$\begin{aligned} &\min_{D,X}\|Y - DX\|_2^2 + \alpha\mathrm{Tr}(XL_g X^T) + \beta\|X\|_1 \\ &\text{s.t. } \|d_i\|^2 \le c, \quad i = 1, \ldots, K \end{aligned} \tag{32}$$

where L_g is the graph Laplacian matrix, and it reflects the locality information of the training samples.

In an ideal condition, we can assume that $Y = DX$. Thus

$$Y = DX \Rightarrow YX^T = DXX^T. \tag{33}$$

If XX^T is a nonsingle value matrix, then

$$YX^T = DXX^T \Rightarrow YX^T(XX^T)^{-1} = D. \tag{34}$$

We defined $\bar{X} = X^T(XX^T)^{-1}$, and then graph Laplacian matrix L was constructed using dictionary D. According to the Graph-SC algorithm, the objective function was defined as follows:

$$\min_{Y,\bar{X}}\|D - Y\bar{X}\|_2^2 + \alpha\mathrm{Tr}(\bar{X}L(\bar{X})^T) + \beta\|\bar{X}\|_1. \tag{35}$$

Therefore, the Graph-SC algorithm can also be regarded as a special form of the LCLE-DL algorithm when the coding coefficient matrix is exactly $X^T(XX^T)^{-1}$.

$$\begin{aligned} \min_{W,V}\|H - WV\|_2^2 &\Leftrightarrow \min_{W,V}\mathrm{Tr}(V^T W^T WV) - \min_{W,V}\mathrm{Tr}(H^T WV) \\ &\Leftrightarrow \min\sum_{i=1}^{C}\left(\begin{array}{l}(v_{(i-1)f+1,1} + \cdots + v_{\mathrm{if},1})^2 + \cdots + (v_{(i-1)f+1,N} + \cdots + v_{\mathrm{if},N})^2 \\ \quad - \sum_{r=(i-1)\times f+1,}^{(i-1)\times f+f}\ \sum_{s=(i-1)\times\frac{N}{C}+1}^{(i-1)\times\frac{N}{C}+\frac{N}{C}} v_{r,s}\end{array}\right) \end{aligned} \tag{31}$$

C. Connection to the LLC Algorithm

In [14], the objective function of the LLC algorithm is

$$\min_X \sum_{i=1}^{N} \|y_i - Dx_i\|^2 + \lambda \|b_i \odot x_i\| \tag{36}$$

where $\odot$ denotes the elementwise multiplication

$$b_i = \exp\left(\frac{\text{dist}(y_i, D)}{\delta}\right)$$

$\text{dist}(y_i, D) = [\text{dist}(y_i, d_1), \ldots, \text{dist}(y_i, d_K)]^T$ and $\text{dist}(y_i, d_j)$ is the Euclidean distance between y_i and d_j, and δ is used for adjusting the weighed decay speed for the locality adaptor. The term $\sum_{i=1}^{N} \|b_i \odot x_i\|$ was rewritten as follows:

$$\begin{aligned}
&\sum_{i=1}^{N} \|b_i \odot x_i\|^2 \\
&\quad = \sum_{i=1}^{N} \|b_{i,1}x_{i,1} + \cdots + b_{i,K}x_{i,K}\|^2 \\
&\quad = \sum_{i=1}^{N}\sum_{j=1}^{K} (b_{i,j})^2 (x_{i,j})^2 + \sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{s=1}^{j} b_{i,j}x_{i,j}b_{i,s}x_{i,s}.
\end{aligned} \tag{37}$$

Since $b_{i,j} = \text{dist}(y_i, d_j)$, (37) shows that the essence of the LLC algorithm is to minimize the weighted coefficients of the coding coefficient matrix.

The locality constrained term of the LCLE-DL algorithm was written as follows:

$$\begin{aligned}
\text{Tr}(X^T L X) &= \sum_{i=1}^{K}\sum_{j=1}^{N} M_{i,j} \|\hat{x}_i - \hat{x}_j\|^2 \\
&= 2\sum_{i=1}^{K}\sum_{j=1}^{K} (x_{i,j})^2 T_{i,i} - 2\sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{s=1}^{N} M_{i,j}(x_{i,s}x_{j,s}).
\end{aligned} \tag{38}$$

Since $\sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{s=1}^{N} M_{i,j}(x_{i,s}x_{j,s}) = \text{Tr}(X^T M X) \geq 0$

$$\min_X(\text{Tr}(X^T L X)) \leq \min\left(\sum_{i=1}^{K}\sum_{j=1}^{K} (x_{i,j})^2 T_{i,i}\right). \tag{39}$$

The essence of the locality constrained term is also to minimize the weighted coefficients of the coding coefficient matrix in the LCLE-DL algorithm. The difference is that the weighted coefficients of the LCLE-DL algorithm is obtained by calculating the similarity of atoms, while the weighted coefficients of the LLC algorithm is obtained by calculating the distance between the atoms and the training samples.

V. Experimental Results and Analyses

In this section, a series of experiments was performed on the CMU PIE face database (PIE) [36], Labeled Faces in the Wild (LFW) database [37], AR face database (AR) [38], Caltech 101 object category database (Caltech 101) [39], and Extended Yale B face database [40]. In order to evaluate the performance of the LCLE-DL algorithm, we compared it with the K-SVD [7], D-KSVD [10], LC-KSVD2 [8], Sparse Representation-based Classification algorithm (SRC) [4], Linear Regression Classification algorithm (LRC) [41], and LLC [14] algorithms. Moreover, we used the whole image approach in these experiments. For the PIE face database, LFW database, AR face database, and Extended Yale B face database, we used the pixels of the whole image as a feature vector. For the Caltech 101 database, we used the spatial pyramid features of the whole image as a feature vector.

Fig. 3. Examples of images from the PIE face database.

Moreover, we also assessed the training time and testing time of the LCLE-DL algorithm and six comparison algorithms on the five databases. The SRC algorithm uses the l1-ls to obtain the representation coefficients of the test samples, so we did not calculate its training time. The LRC algorithm is implemented using the method in [41], so the training time does not need to be calculated. The LLC algorithm obtains the coding coefficients by using the approximated LLC, and the codes are provided by the author.[1] Moreover, the training samples are used as the bases in the LLC algorithm, and it also has no training time. In other words, the SRC, LRC, and LLC algorithms do not learn any dictionary and directly exploit the original training samples to perform classification, so there is no training time. We used the K-SVD box to train the dictionary, and the codes were provided by the author.[2] The codes of the D-KSVD and LC-KSVD2 algorithms were provided by the authors.[3,4] In addition, the K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms use the same classification method presented in Section III-D. The MATLAB codes of this paper can be downloaded from the website (http://www.scholat.com/laizhihui).

A. Experimental Results on the CMU PIE Face Database

The PIE face database consists of 41 386 front-face images of 68 persons, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions. Examples of images from the PIE face database are displayed in Fig. 3.

Experimental Setting: Following [8], sparsity factor $\xi = 30$ was used in the K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms. Moreover, in order to make the comparisons fair, the number of local bases was identical to the sparsity factor ξ in the LLC algorithm. In the following all experiments, parameters k and δ of graph Laplacian matrix L in (3) were set to 1 and 4, respectively. In addition, the first five images of each person were selected to evaluate parameters α, β, and γ, and they were determined by the fivefold cross-validation method. The parameters were selected from $10^{-5}, 10^{-4}, \ldots, 1, \ldots 10^3, 10^4$. α, β, and γ were set to 0.001, 0.1, and 0.01 in this experiment.

[1] http://www.ifp.illinois.edu/~jyang29/LLC.htm.
[2] http://www.cs.technion.ac.il/~elad/software/.
[3] http://www.public.asu.edu/~qzhang53/research.html.
[4] http://www.umiacs.umd.edu/~zhuolin/Resources.html.

TABLE II

AVERAGE RECOGNITION RATES AND COMPUTING TIME ON THE PIE FACE DATABASE

Algorithm	Average recognition rates (%)	Training time (s)	Testing time(s)
LRC(680)	61.6±0.021	-	1.5e-2
LLC	53.7±0.016	-	3.6e-2
SRC(680)	72.1±0.008	-	23.0
K-SVD(680)	72.4±0.010	24.3	2e-3
D-KSVD(680)	71.9±0.008	411	2e-3
LC-KSVD2(680)	72.3±0.009	74.5	2e-3
LCLE-DL(544)	73.9±0.012	-	-
LCLE-DL(680)	75.6±0.009	30.6	2e-3

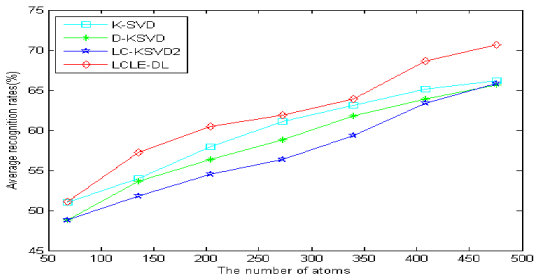

Fig. 4. Average recognition rates with different numbers of atoms on the PIE face database.

Following [42], we chose the five near frontal poses (C05, C07, C09, C27, and C29) and used all the images under different illuminations and expressions, and thus obtained 170 images for each person and every image was normalized to the size of 32×32. We randomly selected ten images of each person (including the first five images) as the training samples and the remaining for testing. The LCLE-DL and six comparison algorithms were repeated ten times. The average recognition rates and computing time are reported in Table II. Moreover, the numbers in the parentheses indicate the number of atoms or training samples, and the symbol $\pm$ denotes the standard deviation of average recognition rates.

In Table II, when the number of atoms was 680, it can be seen that the LCLE-DL algorithm achieved a higher average recognition rate than the six comparison algorithms. In particular, when the number of atoms was smaller than the number of the training samples (for example, 544), the average recognition rate of the LCLE-DL algorithm also outperformed the six comparison algorithms.

Moreover, we compared the average recognition rates of the K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms using different numbers of atoms ($K = 68, 136, \ldots, 408, 476$). The experimental results are displayed in Fig. 4, which shows that the average recognition rates of the K-SVD, D-KSVD LC-KSVD2, and LCLE-DL algorithms increased with an increase in the number of atoms. Moreover, it is obvious that the average recognition rates of the LCLE-DL algorithm outperformed the three dictionary learning algorithms. In addition, it is surprising that the LC-KSVD2 performed slightly worse than the K-SVD and D-KSVD algorithms in most cases.

Fig. 5 shows the optimization process of the LCLE-DL algorithm for $T_{\max} = 50$ iterations using $K = 680$.

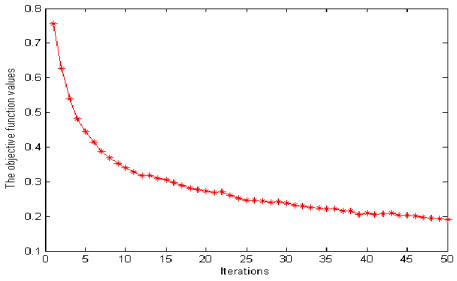

Fig. 5. Variations of the objective function values of the LCLE-DL algorithm with $K = 680$ in the iterations.

Fig. 6. Examples of images from the LFW face database.

TABLE III

AVERAGE RECOGNITION RATES AND COMPUTING TIME ON THE LFW FACE DATABASE

Algorithm	Average recognition rates (%)	Training time(s)	Testing time(s)
LRC(688)	37.1±0.014	-	1.6e-2
LLC	34.8±0.011	-	3.8e-2
SRC(688)	38.1±0.011	-	1.9
K-SVD(688)	32.4±0.020	24.4	2.3e-3
D-KSVD(688)	33.4±0.016	419	2.5e-3
LC-KSVD2(688)	32.2±0.012	40.1	2.1e-3
LCLE-DL(602)	36.8±0.013	-	-
LCLE-DL(688)	38.8±0.009	31.4	2.2e-3

As expected, the objective function values decreased, and thus the LCLE-DL algorithm converged very fast.

B. Experimental Results on the LFW Face Database

The LFW face database contains more than 13 000 images of faces collected from the Web, and they are labeled with the name of the person pictured. The main goal is to study the problem of unconstrained face recognition. There are 1680 images of the persons with two or more distinct photos in the database. Following [43], we selected a subset of the LFW face database consisting of 1215 images of 86 persons, and around 11–20 images for each person. Each image was converted into gray image and was manually cropped and resized to 32×32. Examples of images from the LFW face database are displayed in Fig. 6.

The experiments were performed in the same way as in Section A. The optimal parameters of the LCLE-DL algorithm were $\alpha = 10^{-2}$, $\beta = 10^{-2}$, and $\gamma = 10^{-1}$. We randomly selected eight images of each person (including the first five images) as the training samples and the remaining for testing. We repeated the LCLE-DL and six comparison algorithms ten times. The average recognition rates and computing time are shown in Table III. Moreover, the numbers in the parentheses indicate the number of atoms or training samples, and the symbol $\pm$ denotes the standard deviation of the average recognition rates.

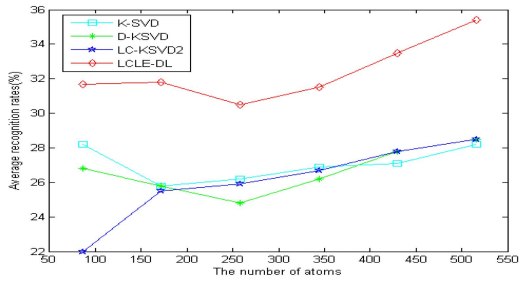

Fig. 7. Average recognition rates with different numbers of atoms on the LFW face database.

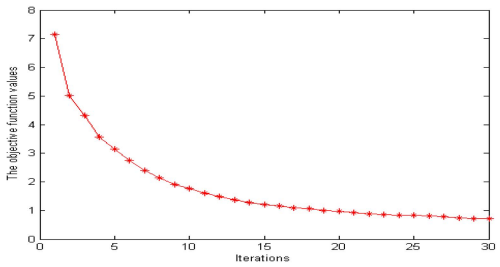

Fig. 8. Variations of the objective function values of the LCLE-DL algorithm with $K = 688$ in the iterations.

As shown in Table III, the average recognition rate of the LCLE-DL algorithm consistently achieved the highest average recognition rate when the number of atoms was 688. Moreover, the average recognition rate of the LLC algorithm was higher than the K-SVD, D-KSVD, and LC-KSVD2 algorithms. However, they all obtained lower average recognition rates than the SRC algorithm. In general, the classification performance of the learned dictionary is higher than those of directly using the original training samples. However, the experimental results indicated that using the label information or locality information cannot always obtain higher average recognition rates than the SRC algorithm.

The average recognition rates of the K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms using different numbers of atoms $(K = 86, 172, \ldots, 430, 516)$ are shown in Fig. 7. It can be seen that the average recognition rates of the LCLE-DL algorithm significantly outperformed the three dictionary learning algorithms as the number of atoms increased.

The convergence of the LCLE-DL algorithm for $T_{\max} = 30$ iterations using $K = 688$ is shown in Fig. 8. As expected, the objective function values decreased very fast and the LCLE-DL algorithm converged very fast.

C. Experimental Results on the AR Face Database

The AR face database contains over 4000 images of 126 persons. There are 26 face images of each person taken during two sessions, and each image is taken under various lighting conditions. Examples of images from the AR face database are displayed in Fig. 9.

We used a subset of the AR face database consisting of 3120 images from 120 persons. The resolution of the AR images was 40×50. For each person, there were 12 images

Fig. 9. Examples of images from the AR face database.

TABLE IV
AVERAGE RECOGNITION RATES AND COMPUTING TIME ON THE AR FACE DATABASE

Algorithm	Average recognition rates(%)	Training time(s)	Testing time(s)
LRC(960)	69.7±0.074	-	3.6e-2
LLC	71.1±0.06	-	1.3e-1
SRC(960)	72.2±0.072	-	10.6
K-SVD(840)	25.5±0.048	-	-
K-SVD(960)	78.8±0.063	85.5	5.4e-3
D-KSVD(840)	69.3±0.056	-	-
D-KSVD(960)	74.1±0.067	2596	5.3e-3
LC-KSVD2(840)	78.5±0.065	-	-
LC-KSVD2(960)	74.2±0.066	223	5.4e-3
LCLE-DL(840)	79.3±0.065	-	-
LCLE-DL(960)	79.4±0.065	94.6	5.2e-3

including sunglasses and scarves. For the training samples, we chose seven neutral images of session 1 and one corrupted image of each person (the first sunglass images at sessions 1 and 2 were not used as the training samples). Thus, a total of 16 test images (seven neutral images at session 2 plus the remaining nine occluded images) were available for testing. Since there were ten corrupted images of each person (two corrupt images were used to evaluate the parameters), the experiments were repeated ten times. Moreover, the experiments were performed as the same way in Section A. The optimal parameters of the LCLE-DL algorithm were $\alpha = 10^{-4}$, $\beta = 10^{-3}$, and $\gamma = 10^{-3}$. The average recognition rates and computing time are reported in Table IV. Moreover, the numbers in the parentheses indicate the number of atoms or training samples, and the symbol $\pm$ denotes the standard deviation of the average recognition rates.

As shown in Table IV, the average recognition rate of the LCLE-DL algorithm was the highest when the number of atoms was 960. Moreover, it is interesting to observe that the average recognition rates of the D-KSVD and LC-KSVD2 algorithms were nearly equal. This is because they both use the label information in the dictionary learning processing although they deal with it in different ways. This also indicates that the label information can play an important role in learning a discriminative dictionary. Since the LCLE-DL algorithm makes use of the label embedding and locality information of the learned dictionary, it can further improve the classification performance.

Fig. 10 shows the average recognition rates of the K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms using different numbers of atoms $(K = 120, 240, \ldots, 600, 720)$. It can be seen that the average recognition rates of the K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms increased, when the number of atoms trended to become large. Moreover, it is obvious that the average recognition rates of the LCLE-DL algorithm outperformed the three dictionary

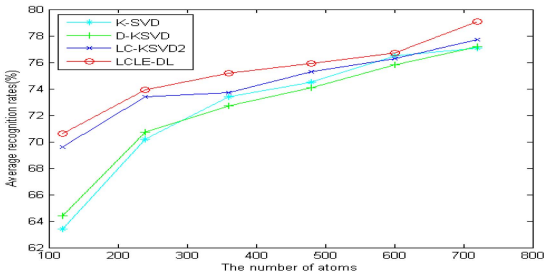

Fig. 10. Average recognition rates with different numbers of atoms on the AR face database.

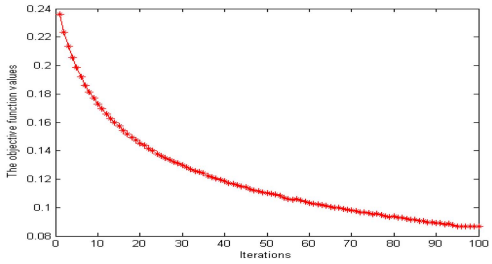

Fig. 11. Variations of the objective function values of the LCLE-DL algorithm with $K = 960$ in the iterations.

Fig. 12. Examples of images from the Caltech 101 database.

learning algorithms. Fig. 11 shows the optimization process of the LCLE-DL algorithm for $T_{\max} = 100$ iterations using $K = 960$. As expected, the objective function values decreased on the whole.

D. Experimental Results on the Caltech 101 Object Category Database

The Caltech 101 object category database consists of 9144 images of 102 classes (101 object classes and a background class). There are around 31–800 images for each class, including animals, vehicles, and flowers. Following [8], the spatial pyramid features were used in the experiments. The spatial pyramid features were extracted based on the Scale-Invariant Feature Transform (SIFT) features with three grids of size 1×1, 2×2, and 4×4. The SIFT descriptors were 16×16 patches, which were densely sampled using a grid with a step size of six pixels. Moreover, the codebook of the spatial pyramid was trained using the standard k-means clustering with $k = 1021$. The spatial pyramid feature was reduced to 3000 dimensions by the Principal Component Analysis methods algorithm. The experiments were performed in the same way as in Section A, and the optimal parameters of the LCLE-DL algorithm were $\alpha = 10^{-5}$, $\beta = 10^{-1}$, and $\gamma = 1$. Examples of images from the Caltech 101 database are displayed in Fig. 12.

TABLE V

AVERAGE RECOGNITION RATES AND COMPUTING TIME ON THE CALTECH 101 DATABASE

Algorithm	Average recognition Rates(%)	Training time(s)	Testing time(s)
LRC(1020)	54.3±0.006	-	6.2e-2
LLC	56.9±0.010	-	1.7e-1
SRC(1020)	57.1±0.008	-	8.7
K-SVD(918)	55.5±0.013	-	-
K-SVD(1020)	23.2±0.139	137	3.4e-3
D-KSVD(918)	58.3±0.009	-	-
D-KSVD(1020)	61.2±0.006	3747	3.3e-3
LC-KSVD2(918)	60.1±0.008	-	-
LC-KSVD2(1020)	60.9±0.050	188	3.7e-3
LCLE-DL(918)	60.2±0.010	-	-
LCLE-DL(1020)	61.6±0.010	132	3.5e-3

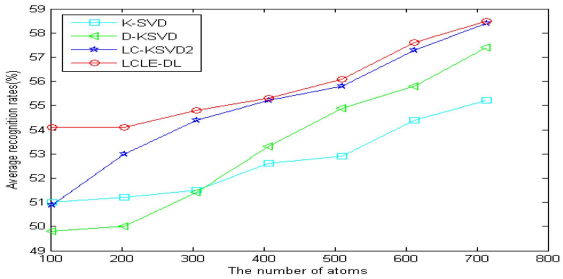

Fig. 13. Average recognition rates with different numbers of atoms on the Caltech 101 database.

We randomly selected ten images (including the first five images) of each class as the training samples and the remaining for testing. The LCLE-DL and six comparison algorithms were performed ten times. The average recognition rates and computing time are reported in Table V. Moreover, the numbers in the parentheses indicate the number of atoms or training samples, and the symbol ± denotes the standard deviation of average recognition rates.

As shown in Table V, the LCLE-DL algorithm achieved a higher average recognition rate than the K-SVD, D-KSVD, LC-KSVD2, and LLC algorithms when the number of atoms was 1020. In particular, when the number of atoms was smaller than the number of training samples, the D-KSVD, LC-KSVD2, and LCLE-DL algorithms achieved higher average recognition rates than the SRC and LRC algorithms.

In addition, the average recognition rates of the K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms using different numbers of atoms $(K = 102, 204, \ldots, 612, 714)$ are shown in Fig. 13. It can be seen that the LCLE-DL algorithm achieved higher average recognition rates than the K-SVD, D-KSVD, and LC-KSVD2 algorithms. Fig. 14 shows the optimization process of the LCLE-DL algorithm for $T_{\max} = 100$ iterations using $K = 1020$. As expected, the objective function values decreased on the whole.

E. Experimental Results on the Extended Yale B Face Database

The Extended Yale B face database is taken under various illumination conditions and expressions, which consists

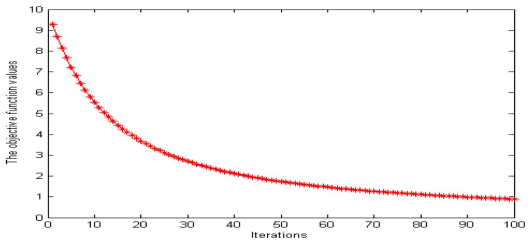

Fig. 14. Variations of the objective function values of the LCLE-DL algorithm with $K = 1020$ in the iterations.

Fig. 15. Examples of images from the Extended Yale B face database.

TABLE VI
AVERAGE RECOGNITION RATES AND COMPUTING TIME ON THE EXTENDED YALE B FACE DATABASE

Algorithm	Average recognition Rates(%)	Training time(s)	Testing time(s)
LRC(760)	92.4±0.008	-	1.8e-2
LLC	88.9±0.010	-	3.8e-2
SRC(760)	95.3±0.005	-	5.6
K-SVD(456)	94.0±0.005	-	-
K-SVD(760)	95.3±0.016	29.7	2.3e-3
D-KSVD(456)	94.3±0.005	-	-
D-KSVD(760)	83.0±0.026	778	2.3e-3
LC-KSVD2(608)	92.9±0.008	-	-
LC-KSVD2(760)	92.7±0.008	92.1	2.3e-3
LCLE-DL(722)	95.4±0.005	-	-
LCLE-DL(760)	95.8±0.005	41.9	2.5e-3

of 2414 front-face images of 38 persons. There are around 59–64 images for each person, and each image was normalized to the size of 32×32. Examples of images from the Extended Yale B face database are displayed in Fig. 15.

The experiments were performed in the same way as in Section A, and the optimal parameters of the LCLE-DL algorithm were $\alpha = 10^{-3}$, $\beta = 10^{-5}$, and $\gamma = 10^{-3}$. We randomly selected 20 images of each person (including the first five images) as the training samples and the remaining for testing. The LCLE-DL and six comparison algorithms were run ten times. The average recognition rates and computing time are reported in Table VI. Moreover, the numbers in the parentheses indicate the number of atoms or training samples, and the symbol $\pm$ denotes the standard deviation of average recognition rates. Table VI shows that the average recognition rates of the LCLE-DL algorithm outperformed the six comparison algorithms.

Moreover, we compared the average recognition rates of the K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms using different numbers of atoms ($K = 38, 76, \ldots, 760$), and the experimental results were displayed in Fig. 16.

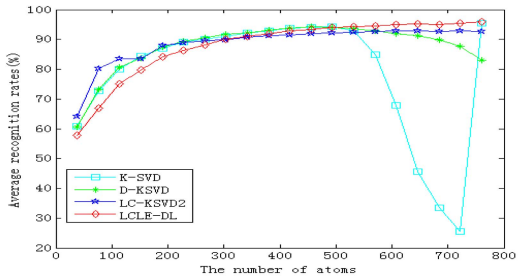

Fig. 16. Average recognition rates with different numbers of atoms on the Extended Yale B face database.

When the number of atoms was increased, the average recognition rates of the LC-KSVD2 and LCLE-DL algorithms increased. However, since the K-SVD algorithm focuses on only the representational power of the learned dictionary without considering its capability for discrimination, it leads to an unstable classification performance. With the increasing number of atoms, the learned dictionary has more redundancy. Thus, the discriminative ability of the learned dictionary may degrade, and then the average recognition rates of the K-SVD algorithm have much variation.

F. Experimental Results on the Single Constraint

In this section, in order to highlight the importance of using two constrained terms than single one, we gave the experimental results by only using the label embedding term or the locality constrained term. When we only used the label embedding term, we denoted the proposed algorithm as the LCLE-DL-V algorithm. The objective function of the LCLE-DL-V algorithm was

$$\min_{D,X} \|Y - DV\|_2^2 + \beta \mathrm{Tr}(V^T U V) + \gamma \|V\|_2^2$$
$$\text{s.t. } \|d_i\|^2 = 1, \quad i = 1, \ldots, K. \tag{40}$$

When we only used the locality constrained term, we denoted the proposed algorithm as the LCLE-DL-X algorithm. The objective function of the LCLE-DL-X algorithm was

$$\min_{D,X,L} \|Y - DX\|_2^2 + \alpha \mathrm{Tr}(X^T L X) + \gamma \|X\|_2^2$$
$$\text{s.t. } \|d_i\|^2 = 1, \quad i = 1, \ldots, K. \tag{41}$$

The experiments were performed on the PIE face database in the same setting as in Section A. Table VII showed the average recognition rates of the LCLE-DL, LCLE-DL-V, and LCLE-DL-X algorithms with different numbers of atoms on the PIE face database. The experimental results showed that the LCLE-DL algorithm achieved higher average recognition rates than the LCLE-DL-V and LCLE-DL-X algorithms on the PIE face database. The main reason was that the dual constraints of the LCLE-DL algorithm ensured that the locality-based and label-based coding coefficients were as close as possible, which can not only transform the discriminative information to the learned dictionary but also endure a certain error. Therefore, it is reasonable to believe that using both label embedding and locality constrained is better than using either of them.

TABLE VII
AVERAGE RECOGNITION RATES OF THE THREE ALGORITHMS ON THE PIE FACE DATABASE

The number of atoms	LCLE-DL (%)	LCLE-DL-V (%)	LCLE-DL-X (%)
68	51.1±0.012	49.0±0.014	49.1±0.011
136	57.2±0.011	55.6±0.011	56.0±0.011
204	60.5±0.011	58.8±0.008	59.2±0.012
272	61.9±0.007	59.8±0.010	60.7±0.011
340	63.9±0.010	62.1±0.008	63.0±0.015
408	68.6±0.009	67.4±0.008	68.0±0.014
476	70.7±0.010	68.8±0.010	70.0±0.010
544	73.9±0.012	70.1±0.009	71.6±0.013
612	74.5±0.010	70.3±0.011	72.3±0.014
680	75.6±0.009	72.4±0.010	73.8±0.012

G. Experimental Analyses

In Sections V-A–V-F, the experimental results on the five image databases were elaborated. A number of interesting points can be drawn as follows.

1) When the number of atoms is equal to the number of training samples, the average recognition rates of the LCLE-DL algorithm outperformed the SRC and LRC algorithms on the five image databases. This demonstrates that the LCLE-DL algorithm can learn a more powerful discriminative dictionary than directly using the original training samples. In addition, when a smaller number of atoms is learned, the LCLE-DL algorithm also achieved higher average recognition rates than the SRC and LRC algorithms on the PIE face database, AR face database, and Caltech 101 database. A smaller number of atoms can significantly reduce the computational cost.
2) The average recognition rates of the LCLE-DL algorithm were higher than the LLC algorithm on the five image databases. The reason is that the LLC algorithm ignores the label information, which may result in the same class training samples with different coding coefficients. Another reason may be that the LLC algorithm uses the distances between the training samples and the atoms to select the k-nearest neighbor atoms for coding. Thus, the true distance relationship is affected by the noise of the training samples. In the LCLE-DL algorithm, the distances among atoms are used to select the k-nearest neighbor atoms for coding, which can reflect the relationship of training samples to some extent. Thus, the potential drawbacks of the LLC algorithm are overcome by the LCLE-DL algorithm.
3) The average recognition rates of the LCLE-DL algorithm were superior to the K-SVD, D-KSVD, and LC-KSVD2 algorithms on the PIE face database, AR face database, LFW database, and Caltech 101 database. This is probably because the D-KSVD and LC-KSVD2 algorithms ignore the locality information of the learned dictionary. Moreover, since the K-SVD algorithm only focuses on the reconstructive ability, it also leads to lower performance in classification. However, the average recognition rates of the LCLE-DL algorithm were worse than the K-SVD, D-KSVD, and LC-KSVD2 algorithms on the Extended Yale B face database when the number of atoms was lower than 456. In particular, the K-SVD and D-KSVD algorithms achieved higher average recognition rates than the LCLE-DL and LC-KSVD2 algorithms. When the number of atoms was larger than 456, the LCLE-DL algorithm achieved a better performance than the three dictionary learning algorithms. However, the average recognition rates of the K-SVD and D-KSVD algorithms dropped in this case. The main reason is that the training samples are relatively larger on the Extended Yale B face database than the other four image databases. This indicates that the reconstruction ability is more important than the geometric properties when the number of training samples is large. However, if the data lie on the nonlinear manifold embedded in a very high-dimensional ambient space, the learned dictionary cannot well reconstruct the training samples and it can degrade the classification performance. The LCLE-DL algorithm overcomes this shortcoming by combining the locality information and label embedding.
4) Experimental results showed that our proposed algorithm had less training time than the D-KSVD and LC-KSVD2 algorithms on the five databases, and had slightly more time than the K-SVD algorithm. The K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms had less testing time than the SRC, LRC, and LLC algorithms on the five databases. Since the K-SVD, D-KSVD, LC-KSVD2, and LCLE-DL algorithms used the same classification method, the testing time of them was nearly equal on each database.

VI. CONCLUSION

We propose a novel dictionary learning algorithm based on the locality and label information of the learned dictionary. The main advantage of the LCLE-DL algorithm is that it can learn a discriminative dictionary, from which a novel graph Laplacian matrix can be further derived. The derived graph Laplacian matrix is used as a local regularization term to encode the representation coefficients. Moreover, the label information of atoms is also used as another regularization term to enforce the strong discriminative property of the learned dictionary. The double reconstruction terms of the LCLE-DL algorithm can lead to the locality reconstruction and label reconstruction being fitted at the same time, which is very helpful for image classification. The proposed dictionary learning algorithm is also more robust to noise than the conventional dictionary learning algorithms by directly using the locality information of the training samples. Experimental results on the five image databases demonstrate that the LCLE-DL algorithm is overall superior to the six state-of-the-art dictionary learning and sparse coding algorithms.

APPENDIX A

Since all class atoms have the same number f in the LCLE-DL algorithm, we can rewrite scaled label matrix U

as follows:

$$U = \begin{bmatrix} U_1 & 0 & \cdots & 0 \\ 0 & U_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & U_C \end{bmatrix} \in \Re^{K \times K} \tag{A-1}$$

where $U_1 = U_2 = \cdots U_C = (1/f) \times O_f \in \Re^{f \times f}$. O_f is a matrix and its elements are all equal to 1. It is obvious that scaled label matrix U is a block-diagonal. Moreover, the block matrix of profile matrix $V^T \in \Re^{N \times K}$ can be written as

$$\begin{aligned} V^T &= \left[\overbrace{\hat{v}_1, \ldots \hat{v}_f}^{1}, \overbrace{\hat{v}_{f+1}, \ldots \hat{v}_{2f}}^{2}, \ldots, \overbrace{\hat{v}_{(C-1)\times f+1} \ldots \hat{v}_K}^{C} \right] \\ &= [\tilde{V}_1, \ldots, \tilde{V}_C] \end{aligned} \tag{A-2}$$

where $\hat{v}_i = [v_{i1}, v_{i2}, \ldots, v_{iN}]^T \in \Re^{1 \times N}$ $(i = 1, 2, \ldots, K)$ and $\tilde{V}_m = [\hat{v}_{(m-1)f+1}, \hat{v}_{(m-1)f+2}, \ldots, \hat{v}_{mf}] \in \Re^{N \times f}$ $(m = 1, 2, \ldots, C)$.

Thus

$$\mathrm{Tr}(V^T U V) = \mathrm{Tr}\left(\frac{1}{f} [\tilde{V}_1 O_f (\tilde{V}_1)^T + \cdots + \tilde{V}_C O_f (\tilde{V}_C)^T] \right). \tag{A-3}$$

Since

$$\mathrm{Tr}(\tilde{V}_m O_f (\tilde{V}_m)^T) = \sum_{j=1}^{N} (v_{(m-1)f+1,j} + \cdots + v_{mf,j})^2. \tag{A-4}$$

Thus

$$\begin{aligned} &\min_V \mathrm{Tr}(V^T U V) \\ &\quad = \min \sum_{m=1}^{C} \sum_{j=1}^{N} (v_{(m-1)f+1,j} + \cdots + v_{mf,j})^2 \\ &\quad = \min \sum_{m=1}^{C} \left(\begin{array}{l} (v_{(m-1)f+1,1} + \cdots + v_{mf,1})^2 \\ \quad + \cdots + (v_{(m-1)f+1,N} + \cdots + v_{mf,N})^2 \end{array} \right). \end{aligned} \tag{A-5}$$

Then, if

$$\begin{gathered} v_{(m-1)f+1,1} = v_{(m-1)f+2,1} = \cdots = v_{mf,1} \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ v_{(m-1)f+1,N} = v_{(m-1)f+2,N} = \cdots = v_{mf,N} \end{gathered}$$

(A-3) can achieve a minimum value.

The vectors

$$\begin{gathered} \hat{v}_{(m-1)f+1} = (v_{(m-1)f+1,1} \cdots v_{(m-1)f+1,N})^T \\ \vdots \qquad\qquad \vdots \\ \hat{v}_{mf} = (v_{mf,1} \cdots v_{mf,N})^T \end{gathered}$$

are just the profiles of the mth class atoms $d_{(m-1)f+1}, \ldots, d_{mf}$. Therefore, it indicates that the label embedding constraint encourages the same class atoms to have similar profiles.

APPENDIX B

According to [29] and [30], the basic model of dictionary learning can be rewritten as follows:

$$Y = d_1(\hat{x}_1)^T + \cdots d_i(\hat{x}_i)^T + \cdots d_j(\hat{x}_j)^T + \cdots + d_K(\hat{x}_K)^T \tag{B-1}$$

where $\hat{x}_m, (m = 1, \ldots, K)$ is the profile of atom d_m. Then, we can define

$$E = \sum_{m=1(m \neq i,j)}^{K} d_m(\hat{x}_m)^T. \tag{B-2}$$

Thus, (B-1) becomes

$$Y = E + d_i(\hat{x}_i)^T + d_j(\hat{x}_j)^T. \tag{B-3}$$

In order to obtain profile $\hat{x}_i$, (B-3) becomes

$$(d_i)^T (Y - E - d_j(\hat{x}_j)^T) = (d_i)^T d_i(\hat{x}_i)^T. \tag{B-4}$$

Thus

$$(\hat{x}_i)^T = ((d_i)^T d_i)^{-1} (d_i)^T (Y - E - d_j(\hat{x}_j)^T). \tag{B-5}$$

Similarly

$$(\hat{x}_j)^T = ((d_j)^T d_j)^{-1} (d_j)^T (Y - E - d_i(\hat{x}_i)^T). \tag{B-6}$$

For the sake of simplicity, we can define

$$\psi_j = ((d_j)^T d_j)^{-1} (d_j)^T. \tag{B-7}$$

Then

$$\begin{aligned} &(\hat{x}_i)^T - (\hat{x}_j)^T \\ &\quad = \psi_i (Y - E - d_j(\hat{x}_j)^T) - \psi_j (Y - E - d_i(\hat{x}_i)^T) \\ &\quad = (\psi_i - \psi_j)(Y - E) + \psi_j d_i(\hat{x}_i)^T - \psi_i d_j(\hat{x}_j)^T. \end{aligned} \tag{B-8}$$

Thus, if $\hat{x}_j = \hat{x}_i$, (B-8) becomes

$$(\psi_i - \psi_j)(Y - E) + (\psi_j d_i - \psi_i d_j)(\hat{x}_j)^T = 0. \tag{B-9}$$

Since $(Y - E) \neq 0$ and $\hat{x}_j \neq 0$, $d_i = d_j$. Therefore, it demonstrates that the similar profiles can encourage the corresponding atoms to be similar.

REFERENCES

[1] I. Tosic and P. Frossard, "Dictionary learning: What is the right representation for my signal?" *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.

[2] M. Elad, "Sparse and redundant representation modeling—What next?" *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 922–928, Dec. 2012.

[3] H. Cheng, Z. Liu, L. Yang, and X. Chen, "Sparse representation and learning in visual recognition: Theory and applications," *Signal Process.*, vol. 93, no. 6, pp. 1408–1426, Jun. 2013.

[4] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[5] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1255–1262, Sep. 2011.

[6] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Two-stage nonnegative sparse representation for large-scale face recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 35–46, Jan. 2013.

[7] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[8] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proc. IEEE Conf. CVPR*, Providence, RI, USA, Jun. 2011, pp. 1697–1704.

[9] D.-S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *Proc. IEEE Conf. CVPR*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[10] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Conf. CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 2691–2698.

[11] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[12] 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, Dec. 2000.

[13] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Proc. Adv. NIPS*, Vancouver, BC, Canada, 2009, pp. 2223–2231.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 3360–3367.

[15] C.-P. Wei, Y.-W. Chao, Y.-R. Yeh, and Y.-C. F. Wang, "Locality-sensitive dictionary learning for sparse representation based classification," *Pattern Recognit.*, vol. 46, no. 5, pp. 1277–1287, May 2013.

[16] L. Shen, S. Wang, G. Sun, S. Jiang, and Q. Huang, "Multi-level discriminative dictionary learning towards hierarchical visual categorization," in *Proc. IEEE Conf. CVPR*, Portland, OR, USA, Jun. 2013, pp. 383–390.

[17] Y. Chen, H. Bao, and X. He, "Non-negative local coordinate factorization for image representation," in *Proc. IEEE Conf. CVPR*, Providence, RI, USA, Jun. 2011, pp. 569–574.

[18] D. Arpit, G. Srivastava, and Y. Fu, "Locality-constrained low rank coding for face recognition," in *Proc. 21st ICPR*, Tsukuba, Japan, Nov. 2012, pp. 1687–1690.

[19] Y. Zhou and K. E. Barner, "Locality constrained dictionary learning for nonlinear dimensionality reduction," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 335–338, Apr. 2013.

[20] S. Gao, I. W. Tsang, L.-T. Chia, and P. Zhao, "Local features are not lonely—Laplacian sparse coding for image classification," in *Proc. IEEE Conf. CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 3555–3561.

[21] M. Zheng *et al.*, "Graph regularized sparse coding for image representation," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1327–1336, May 2011.

[22] K. N. Ramamurthy, J. J. Thiagarajan, P. Sattigeri, and A. Spanias, "Learning dictionaries with graph embedding constraints," in *Proc. ASILOMAR*, Pacific Grove, CA, USA, Nov. 2012, pp. 1974–1978.

[23] L. Jing, M. K. Ng, and T. Zeng, "Dictionary learning-based subspace structure identification in spectral clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1188–1199, Aug. 2013.

[24] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE Conf. CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 3501–3508.

[25] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Learning dictionaries with bounded self-coherence," *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 861–864, Dec. 2012.

[26] T. Lin, S. Liu, and H. Zha, "Incoherent dictionary learning for sparse representation," in *Proc. 21st ICPR*, Tsukuba, Japan, Nov. 2012, pp. 1237–1240.

[27] S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 623–634, Feb. 2014.

[28] A. Shaban, H. R. Rabiee, M. Farajtabar, and M. Ghazvininejad, "From local similarity to global coding: An application to image classification," in *Proc. IEEE Conf. CVPR*, Portland, OR, USA, Jun. 2013, pp. 2794–2801.

[29] C. Lu, J. Shi, and J. Jia, "Scale adaptive dictionary learning," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 837–847, Feb. 2014.

[30] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Learning overcomplete dictionaries based on atom-by-atom updating," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 883–891, Feb. 2014.

[31] Y. Zhang, Z. Jiang, and L. S. Davis, "Learning structured low-rank representations for image classification," in *Proc. IEEE Conf. CVPR*, Portland, OR, USA, Jun. 2013, pp. 676–683.

[32] J. Ye, Z. Zhao, and M. Wu, "Discriminative K-means for clustering," in *Proc. Adv. NIPS*, Vancouver, BC, Canada, 2007, pp. 1649–1656.

[33] G. Liu, Z. Liu, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[34] X. Zhu and J. Lafferty, "Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning," in *Proc. 22nd ICML*, 2005, pp. 1052–1059.

[35] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[36] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.

[37] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," College Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007.

[38] A. Martínez and R. Benavente, "The AR face database," Comput. Vis. Center, Barcelona, Spain, Tech. Rep. 24, Jun. 1998.

[39] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. CVPRW*, New York, NY, USA, Jun. 2004, p. 178.

[40] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[41] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, Nov. 2010.

[42] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal Laplacianfaces for face recognition," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3608–3614, Nov. 2006.

[43] S.-J. Wang, J. Yang, M.-F. Sun, X.-J. Peng, M.-M. Sun, and C.-G. Zhou, "Sparse tensor discriminant color space for face verification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 6, pp. 876–888, Jun. 2012.

Zhengming Li received the B.S. degree in computer science and technology from Henan Normal University, Xinxiang, China, in 2004, and the M.S. degree in computer application technology from Southwest University, Chongqing, China, in 2007. He is currently pursuing the Ph.D. degree in computer science and technology with the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China.

His current research interests include pattern recognition and biometrics.

Zhihui Lai received the B.S. degree in mathematics from South China Normal University, Guangzhou, China, in 2002, the M.S. degree from Jinan University, Guangzhou, in 2007, and the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, China, in 2011.

He has been a Research Associate, Post-Doctoral Fellow, and Research Fellow with The Hong Kong Polytechnic University, Hong Kong, since 2010. His current research interests include face recognition, image processing and content-based image retrieval, pattern recognition, compressive sense, human vision modelization, and applications in the fields of intelligent robot research.

Yong Xu (M'06–SM'15) received the B.S. and M.S. degrees from the Air Force Institute of Meteorology, Nanjing, China, in 1994 and 1997, respectively, and the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, in 2005.

Currently, he is a professor in the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. His current research interests include pattern recognition, biometrics, bioinformatics, machine learning, image processing, and video analysis.

Jian Yang received the B.S. degree in mathematics from Xuzhou Normal University, Xuzhou, China, in 1995, the M.S. degree in applied mathematics from Changsha Railway University, Changsha, China, in 1998, and the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002.

He was a Post-Doctoral Researcher with the University of Zaragoza, Zaragoza, Spain, in 2003. He was a Post-Doctoral Fellow with the Biometrics Centre, The Hong Kong Polytechnic University, Hong Kong, from 2004 to 2006, and the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA, from 2006 to 2007. He is currently a Professor with the School of Computer Science and Technology, NUST. He has authored over 50 scientific papers in pattern recognition and computer vision. His current research interests include pattern recognition, computer vision, and machine learning.

Prof. Yang received the RyC Program Research Fellowship by the Spanish Ministry of Science and Technology in 2003.

David Zhang (F'08) received the Degree in computer science from Peking University, Beijing, China, the M.Sc. degree in computer science and the Ph.D. degree from the Harbin Institute of Technology (HIT), Shenzhen, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 1994.

He was a Post-Doctoral Fellow with Tsinghua University, Beijing, and then an Associate Professor with Academia Sinica, Beijing, from 1986 to 1988. He is currently the Head of the Department of Computing, and a Chair Professor with The Hong Kong Polytechnic University, Hong Kong. He also serves as the Visiting Chair Professor with Tsinghua University, and an Adjunct Professor with Peking University, Shanghai Jiao Tong University, Shanghai, China, HIT, and the University of Waterloo. He has authored over ten books and 200 journal papers.

Prof. Zhang is a Croucher Senior Research Fellow, Distinguished Speaker of the IEEE Computer Society, and fellow of the International Association for Pattern Recognition. He is the Founder and Editor-in-Chief of the *International Journal of Image and Graphics*, a Book Editor of *International Series on Biometrics* (Springer), an Organizer of the International Conference on Biometrics Authentication, and an Associate Editor of more than ten international journals, including the IEEE TRANSACTIONS and *Pattern Recognition*.