10-2016

Bilevel model-based discriminative dictionary learning for recognition

Pan ZHOU
Singapore Management University, panzhou@smu.edu.sg

Chao ZHANG

LIN Zhouchen

Bilevel Model-Based Discriminative Dictionary Learning for Recognition

Pan Zhou, Chao Zhang, *Member, IEEE*, and Zhouchen Lin, *Senior Member, IEEE*

***Abstract*— Most supervised dictionary learning methods optimize the combinations of reconstruction error, sparsity prior, and discriminative terms. Thus, the learnt dictionaries may not be optimal for recognition tasks. Also, the sparse codes learning models in the training and the testing phases are inconsistent. Besides, without utilizing the intrinsic data structure, many dictionary learning methods only employ the ℓ_0 or ℓ_1 norm to encode each datum independently, limiting the performance of the learnt dictionaries. We present a novel bilevel model-based discriminative dictionary learning method for recognition tasks. The upper level directly minimizes the classification error, while the lower level uses the sparsity term and the Laplacian term to characterize the intrinsic data structure. The lower level is subordinate to the upper level. Therefore, our model achieves an overall optimality for recognition in that the learnt dictionary is directly tailored for recognition. Moreover, the sparse codes learning models in the training and the testing phases can be the same. We further propose a novel method to solve our bilevel optimization problem. It first replaces the lower level with its Karush–Kuhn–Tucker conditions and then applies the alternating direction method of multipliers to solve the equivalent problem. Extensive experiments demonstrate the effectiveness and robustness of our method.**

***Index Terms*— Sparse representation, dictionary learning, bilevel optimization, recognition, alternating direction method.**

I. Introduction

SPARSE representation has been widely used in signal processing and computer vision, such as signal reconstruction [1], image denosing [2], and recognition [3]–[5], yielding state-of-the-art performance. Its main idea is to represent a signal/sample by a linear combination of a few atoms from a learnt dictionary. Thus, the dictionary quality is an important factor in sparse representation based methods. For example, sparse representation classification (SRC) [3] directly uses the training data as the dictionary and achieves great success in face recognition. Unfortunately, SRC breaks down when the training data are wildly corrupted, because using the noisy data as the dictionary distorts the structure of data [6]–[9]. To resolve this issue, many dictionary learning methods have been proposed. Based upon whether utilizing supervised information in the training phase, one can roughly divide these dictionary learning methods into two kinds: unsupervised methods and supervised ones.

Among unsupervised methods, the method of optimal directions (MOD) [10] and KSVD [1] are classical. Note that these two methods solve the same dictionary model. They only differ in the optimization methods. At each iteration, MOD uses the orthogonal matching pursuit (OMP) algorithm [11] to find a sparse representation and updates the dictionary by solving a least squares problem, while KSVD updates the representation and the dictionary with singular value decomposition (SVD) to accelerate convergence. There are also other performance-impressive methods, such as [12]–[14]. Unsupervised methods construct a dictionary by minimizing the reconstruction error of original samples. Such methods have achieved promising results in signal representation and reconstruction and have also been used for other purposes, such as recognition [3], [15]–[18].

Supervised dictionary learning methods exploit the class labels of the training data, thus can obtain better classification performance than unsupervised methods. In [4], Zhang et al. propose a discriminative KSVD (D-KSVD) dictionary learning method. They consider not only the reconstruction error but also the classification error in their model and utilize KSVD to solve their model. Jiang et al. [5] present a label consistent KSVD (LC-KSVD) dictionary learning method. They explicitly incorporate a label consistency constraint, called the discriminative sparse-code error, and an optimal classification performance criterion into the objective function and solve their model with the KSVD algorithm. Similarly, Mairal et al. [19] and Lian et al. [20] consider the logistic loss function in their models, while Yange et al. [21] and Wang et al. [22] adopt the hinge loss function as the discriminative criterion. There are also other discriminative criteria to guide discriminative dictionary learning. Yang et al. [8] and Zhou et al. [23] propose two Fisher discrimination based dictionary learning approaches. Their models encourage the sparse representation coefficients to have small within-class scatter but large between-class scatter. Supervised methods

Manuscript received April 26, 2016; revised September 15, 2016; accepted October 23, 2016. Date of publication October 30, 2016; date of current version January 20, 2017. The work of P. Zhou and C. Zhang was supported in part by the National Key Basic Research Project of China (973 Program) under Grant 2015CB352303 and in part by the National Nature Science Foundation of China under Grant 61671027 and Grant 61071156. The work of Z. Lin was supported in part by the 973 Program of China under Grant 2015CB352502 and in part by the NSF of China under Grant 61625301 and Grant 61231002. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Abd-Krim Karim Seghouane. *(Corresponding author: Chao Zhang.)*

P. Zhou is with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: pzhou@pku.edu.cn).

C. Zhang and Z. Lin are with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University, Beijing, China, and also with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: chzhang@cis.pku.edu.cn; zlin@pku.edu.cn).

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/TIP.2016.2623487

usually incorporate discriminative terms into the objective functions directly to learn discriminative dictionaries.

However, there are three drawbacks in the aforementioned methods. Firstly, unsupervised methods, without utilizing supervised discriminative information, learn dictionaries only by minimizing the reconstruction error of original samples. Minimizing the reconstruction error may not be closely related to the recognition task that follows. Indeed, recent works [4], [5], [8], [19]–[24] all indicate that supervised dictionary learning methods can yield higher-quality dictionaries and achieve better performance in recognition tasks. Secondly, almost all the aforementioned supervised methods minimize the combinations of the reconstruction error and the classification error (or other discriminative terms), rather than the final goal of discriminative dictionary learning, i.e., the classification error. So the classification using the learnt dictionaries may not be optimal. Another side effect is that the problems for computing the sparse codes in the training and the testing phases have to be different, making the models inconsistent. Finally, most of the unsupervised and supervised dictionary learning methods only employ the ℓ_0 or ℓ_1 norm as the sparsity constraint for dictionary learning. As a result, each sample is encoded independently. Such a mechanism may not take advantage of the structure information of data sufficiently. Actually, the authors of [7], [9], and [25]–[28] all point out that given a dictionary the nonzero coefficients of samples from the same class should cluster, such that they accord with the clustering in the sample space. Thus, seeking the sparsest representation of a sample might not be the best criterion. So the authors of [25]–[28] all employ mixed norms (such as ℓ_q/ℓ_1) as the sparsity criteria to encourage group sparsity of representation. Zhang et al. [7] propose a low-rank representation based dictionary learning (LRRDL) method to capture the data structure. Later they introduce an ideal coding based regularization term into the LRRDL model to learn a structured low-rank dictionary. This new method is called the SLRRDL method. Sun et al. [9] construct a class-specific dictionary by adding a weighted group sparse constraint. Compared with the methods that each sample is encoded independently, these methods can utilize data structure information more sufficiently, hence can achieve better performance in recognition tasks.

In this paper, aiming at overcoming the above three drawbacks, we propose a novel bilevel model based discriminative dictionary learning (BMDDL) method for recognition tasks. Unlike other supervised dictionary learning methods that optimize the combinations of the classification error and other criteria (such as reconstruction error and sparsity constraint), BMDDL directly minimizes the classification error. It is a bilevel optimization model. The upper level aims at minimizing the classification loss, while the lower level aims at characterizing the intrinsic data structure. The objective of lower level is subordinate to that of the upper level. By this way, the dictionary is learnt to minimize the classification error directly. In addition, the problems for computing the sparse codes in the training and the testing phases can be the same. So our model is consistent. What is more, in the lower level we use the Laplacian regularization and sparsity penalty to encourage group sparsity of representation. Therefore, the lower level encourages samples from the same class to have similar sparse codes and those from different classes to have dissimilar sparse representations. Such a mechanism leads to a high quality learnt dictionary, which encourages the representations of samples to preserve the geometric structure within data. At last, we propose a novel method to solve our bilevel model. As far as we know, bilevel model based dictionary learning methods usually use the stochastic subgradient method [29] to solve their models [30], [31]. We replace the lower level problem with its Karush-Kuhn-Tucker (KKT) conditions, which are equality and inequality constraints. Thus we can use the alternating direction method (ADM) of multipliers [32] to solve this equivalent model. The advantage of our approach is that in each iteration we do not have to solve the lower level problem exactly in order to obtain a subgradient of the upper level objective function. Moreover, it is unnecessary to deduce the (sub)gradient via implicit differentiation, which is rather complex when the lower level objective function is non-differentiable, e.g., involving the ℓ_1 norm for sparsity.

Although Yang et al. [31], Mairal et al. [30], Tao et al. [33], and Lobel et al. [34] also use bilevel models to learn discriminative dictionaries, our method is different from theirs. Firstly, the model by Yang et al. [31] is for learning two dictionaries that couple two signal spaces. Its upper level is to minimize the difference between the sparse codes in the two spaces. The model is unsupervised and targets on image superresolution, not recognition tasks. The model by Tao et al. [33] is for semantic segmentation. The upper level minimizes the conditional random field energy function, which is usually used for semantic segmentation, rather than image classification. The task-driven dictionary learning (TDDL) model by Mairal et al. [30] is for recognition as its upper level optimizes the classification error. In [30] and [33], the lower level considers the reconstruction error and the sparsity of representation, but it only adopts the ℓ_1 norm as the sparsity constraint. No structure information of data is considered. By comparison, our model employs the Laplacian regularization to preserve the data structure. Lobel et al. [34] also propose a bilevel model aiming at learning more compact representations for recognition tasks. Its upper level minimizes the combination of the loss function of a linear SVM and the regularization on dictionary. Thus, it also has the second drawback we mentioned previously that the learnt dictionary is optimal for the combination, rather than the classification loss. Accordingly, the dictionary may not be the most discriminative for recognition tasks. Its lower level uses the max-pooling operator to select a few visual dictionary words to construct more compact features. While our upper level only minimizes the classification loss of a linear classifier, which can avoid the second drawback, and our lower level adopts the ℓ_1 norm and the Laplacian regularization to encourage group sparsity. Secondly, [30], [31], and [33] all use the stochastic subgradient method to solve their models. At each iteration, in order to obtain a good descent direction (subgradient), they need to solve a LASSO problem in the lower level at reasonably

good numerical precision, which is computationally expensive when the scale of a dictionary is large [5], [12], [35]. Besides, they use the subgradient of the upper level objective to update the dictionary, which is known to be very slow. By comparison, our optimization method utilizes the KKT conditions to transform the lower level problem into equality and inequality constraints. The equivalent problem can be solved by ADM [32]. In this way, the lower level problem needs not be solved at high precision in each iteration. The KKT conditions only need to be met when convergence. Therefore, our method could be much faster. Lobel et al. [34] use an alternating minimization algorithm based on the CCCP algorithm [36], which is designed for unconstraint optimization problems whose objective is decomposed as the sum of a convex and a concave term. Thus, the applicability of this optimization method is limited.

Note that Laplacian regularization has been utilized in dictionary learning, e.g., Gao et al. [17] present a Laplacian sparse coding (LSC) method. However, LSC is a unilevel optimization model and is unsupervised, while ours is a bilevel model and supervised. Guo et al. [18] propose a pairwise constraint based discriminative dictionary learning method, named DDL-PC. They also incorporate a Laplacian term with a linear classifier to jointly learn a discriminative dictionary and a classifier. However, their model is unilevel, which cannot avoid the second drawback we mentioned above, i.e., non-optimality for classification and model inconsistency. We will discuss the differences between unilevel models and bilevel models in more detail in Section IV. Another difference is that in the testing phase, Guo et al. [18] solve a LASSO problem to compute the sparse codes of testing samples, while we further consider the data structure and solve the lower level optimization problem, i.e., problem (22), to compute the sparse representations of testing samples.

In summary, our main contributions include:

1) We propose a bilevel model for simultaneous discriminative dictionary learning and data classification. The upper level directly minimizes the classification error, while the lower level aims at characterizing the intrinsic data structure. Our model achieves an overall optimality in that the dictionary learning is directly connected to recognition. Moreover, our model is consistent. Namely, the problems for computing the sparse codes in the training and the testing phases can be the same.
2) While most of dictionary learning methods only adopt the ℓ_0 or ℓ_1 norm as the sparsity constraint for dictionary learning and encode each sample independently, which ignores the data structure information, our method employs the supervised Laplacian regularization to preserve the intrinsic data structure.
3) Unlike other bilevel model based dictionary learning methods that employ the stochastic subgradient method to solve their models, we propose a novel method to solve our bilevel model. We utilize the KKT conditions to transform the lower level problem into equality and inequality constraints, then apply ADM to solve the equivalent model.

TABLE I
SUMMARY OF NOTATIONS FREQUENTLY USED IN THIS PAPER

Notation	Meaning
capital letter	A matrix. Especially, I is the identity matrix.
M^T	Transpose of matrix M.
M_{ij}	The (i,j)th entry of matrix M.
M_i	The ith column of matrix M.
$M^\dagger$	Moore-Penrose pseudo-inverse of matrix M.
$\|\|\cdot\|\|_0$	Number of nonzero entries.
$\|\|\cdot\|\|_1$	$\|\|M\|\|_1 = \sum_{i,j} \|M_{ij}\|$.
$\|\|\cdot\|\|_F$	Frobenious norm, $\|\|M\|\|_F = \sqrt{\sum_{i,j} M_{ij}^2}$.
$\|\|\cdot\|\|_2$	Vector Euclidean norm, $\|\|x\|\| = \sqrt{\sum_i x_i^2}$.
$\odot$	$C = A \odot B$, where $C_{ij} = A_{ij} B_{ij}$.
$\oslash$	$C = A \oslash B$, where $C_{ij} = A_{ij} / B_{ij}$.
tr$(\cdot)$	Sum of the diagonal entries of a matrix.

Extensive experimental results demonstrate the advantages of our method.

The remainder of this paper is organized as follows. Section II briefly reviews related work on the existing dictionary learning methods. In Section III, we present our bilevel model based discriminative dictionary learning (BMDDL) method. We also present how to utilize the KKT conditions and ADM to solve our bilevel model. In Section IV, we compare unilevel models with bilevel models and argue for the advantages of bilevel models. Section V presents experimental results and analysis. Finally, Section VI concludes the paper and discusses future work.

II. RELATED WORK

Since the existing dictionary learning methods can be roughly divided into unsupervised and supervised ones, we will briefly introduce these two kinds of methods in turn in this section. For brevity, we summarize some frequently used notations in Table I. Suppose that $Y = [Y_1, \cdots, Y_n] \in \mathbb{R}^{d \times n}$ is the data matrix, in which d is the feature dimension and n is the number of samples. $D \in \mathbb{R}^{d \times k}$ is the dictionary we want to learn, in which k is the number of atoms in the dictionary. $A = [A_1, \cdots, A_n] \in \mathbb{R}^{k \times n}$ is the representation of the feature matrix Y under the dictionary D, where A_i corresponds to the ith sample Y_i.

A. Unsupervised Dictionary Learning

Unsupervised dictionary learning methods usually minimize the combinations of the reconstruction error and the sparsity of the learnt representation. A typical model is:

$$\min_{D,A} \|Y - DA\|_F^2, \text{ s.t. } \|D_i\|_2^2 \le 1, \ \forall i \in \{1, 2, \cdots, k\}, \\ \|A_j\|_0 \le T, \ \forall j \in \{1, 2, \cdots, n\}, \quad (1)$$

where the term $\|Y - DA\|_F^2$ is the reconstruction error. $\|A_j\|_0 \le T$ means that the jth sample has fewer than T nonzero entries in its representation. MOD [10] and KSVD [1] learn a dictionary by solving problem (1). Some unsupervised methods also consider discriminative terms in their models. Besides, the ℓ_0 norm is often approximated by the ℓ_1 norm

TABLE II
SUMMARY OF DISCRIMINATIVE TERMS IN UNSUPERVISED AND SUPERVISED DICTIONARY LEARNING METHODS

Unsupervised	Discriminative term
LSC [17]	$F = \text{tr}(ALA^T)$, where L is the Laplacian matrix.
Supervised	**Discriminative term**
D-KSVD [4]	$F = \|H - WA\|_F^2 + \lambda\|W\|_F^2$, where H is the label matrix and W is the classifier parameter matrix.
LC-KSVD1 [5]	$F = \|Q - BA\|_F^2$, where Q is the ideal sparse codes matrix and B is a linear transformation matrix.
LC-KSVD2 [5]	$F = \|H - WA\|_F^2 + \lambda\|Q - BA\|_F^2$, where H and W have the same meanings as those in D-KSVD.
DDL-PC [18]	$F = \|H - WA\|_F^2 + \lambda\|W\|_F^2 + \gamma\text{tr}(ALA^T)$, where H and W have the same meanings as those in D-KSVD, and L is the Laplacian matrix.
Mairal [19]	$F = \sum_i \log(1 + e^{h_i w^T A_i}) + \lambda\|w\|_2^2$, where h_i is the label of the ith sample and w is the parameter vector.
Zhou [23]	$F = \text{tr}(S_W(A) - S_B(A))$, where $S_W(A)$ and $S_B(A)$ are the within-class scatter and the between-class scatter matrices, respectively.

in order to make the models more easily solvable. A general model can be written as:

$$\begin{aligned} &\min_{D,A,S} \quad \|Y - DA\|_F^2 + \alpha\|A\|_1 + \beta F(D, A, S), \\ &\text{s.t.} \quad \|D_i\|_2^2 \le 1, \ \forall i \in \{1, 2, \cdots, k\}, \end{aligned} \tag{2}$$

where $\|A\|_1$ is a sparse penalty term, $F(D, A, S)$ is a general unsupervised discriminative term, and α and β are two positive parameters controlling the relative contribution of the corresponding terms, respectively. Since this kind of methods construct the discriminative term $F(D, A, S)$ in an unsupervised way, these methods belong to the unsupervised category. The discriminative term of LSC [17] can be found in Table II, where its Laplacian matrix is computed from histogram intersection.

B. Supervised Dictionary Learning

Based on unsupervised dictionary learning methods, most supervised methods directly add discriminative terms to the objective functions of unsupervised methods. So a general supervised dictionary learning model can also be formulated as (2), where $F(D, A, S)$ is a general supervised discriminative term. Many supervised dictionary learning methods, such as [4], [5], [8], and [18]–[23], can be formulated as the above dictionary learning model (2). The discriminative terms of D-KSVD [4], LC-KSVD [5], DDL-PC [18], [19], and [23] are summarized in Table II. Note that LC-KSVD has two versions, LC-KSVD1 and LC-KSVD2. Please refer to [5]. With the class labels of the training data available, supervised dictionary learning methods exploit the class discriminative information and obtain better classification performance than unsupervised methods in most cases. However, since their objective functions consist of the reconstruction error and other discriminative terms, the classification error may not be minimized. Therefore, the learnt dictionary may not be the most discriminative one for recognition tasks. Besides, the ℓ_1 norm cannot well capture the data structure. In [30], a bilevel model based dictionary learning method is proposed. But it also only employs the ℓ_1 norm and encodes each sample independently.

In either kinds of dictionary learning methods, when computing the sparse code for a testing sample, the discriminative term $F(D, A, S)$ has to be dropped. So their models are inconsistent.

III. BILEVEL MODEL BASED DISCRIMINATIVE DICTIONARY LEARNING

In this section, we first present our bilevel model for discriminative dictionary learning. Then we introduce a novel method to solve our bilevel optimization problem. Finally, we summarize our framework for recognition tasks.

A. Model for Discriminative Dictionary Learning

We propose a bilevel model for recognition-driven discriminative dictionary learning. In a recognition task, minimizing the classification error is the ultimate goal. Accordingly, in our model the upper level feeds the representation A_i of the ith sample Y_i into a classifier $f(A_i, W)$ and directly minimizes the classification loss. This goal is primary. The lower level tries to capture the data structure and this goal is secondary. Actually, in most cases the high-dimensional sample points across multiple classes lie in multiple low-dimensional subspaces, and samples in the same class should cluster together as a low-dimensional subspace whose intrinsic dimension is often much smaller than the data dimension. Intuitively, given a dictionary the nonzero coefficients of samples from the same class should also cluster, which can be promoted by group sparsity. To this end, we adopt the combination of the sparsity term and the Laplacian discriminative term to encourage group sparsity of representation. It should be pointed out that we construct the Laplacian matrix in a supervised way. So it can well preserve the data structure even if there exists noise in the data. In this way, the lower level can utilize the intrinsic data structure to optimize for the discrimination capability of the representations with respect to a given dictionary. Such a framework leads to a better recognition-driven dictionary. Our model can be formulated as follows:

$$\begin{aligned} &\min_{W,D} \quad \sum_{i=1}^{n} \varphi(h_i, f(A_i, W)) + \lambda\|W\|_F^2, \\ &\text{s.t.} \quad A = \arg\min_A \frac{1}{2}\|Y - DA\|_F^2 + \alpha\|A\|_1 + \frac{\beta}{2}\text{tr}\left(ALA^T\right), \\ &\qquad \|D_i\|_2^2 \le 1, \ \forall i \in \{1, 2, \cdots, k\}, \end{aligned} \tag{3}$$

where $A_i \in \mathbb{R}^d$ is the representation of the ith sample and W is the parameter matrix of classifier $f(\cdot, W)$. h_i is the 0-1 binary label vector of the ith sample, where the position of 1 indicates the class of Y_i. φ is a classification loss function. $\|A\|_1$ is a sparse penalty term. $L \in \mathbb{R}^{n \times n}$ is the Laplacian matrix of the data matrix Y (please refer to Eq. (30) for constructing L). $\text{tr}\left(ALA^T\right)$ is the Laplacian term, which encourages samples from the same class to have similar sparse codes and those from different classes to have dissimilar sparse representations. λ, α, and β are three regularization parameters.

In this paper, we use a linear predictive classifier $f(x, W) = Wx$ and a quadratic loss function. Actually, this is the multivariate ridge regression [37]. For other classifiers, the resulting optimization problem could still be solved but will be much more involved. Then the optimization problem (3) can be written as follows:

$$\begin{aligned} \min_{W,D} \quad & \|H - WA\|_F^2 + \lambda\|W\|_F^2, \\ \text{s.t.} \quad & A = \arg\min_A \frac{1}{2}\|Y - DA\|_F^2 + \alpha\|A\|_1 + \frac{\beta}{2}\text{tr}\left(ALA^T\right), \\ & \|D_i\|_2^2 \le 1, \ \forall i \in \{1, 2, \cdots, k\}, \end{aligned} \tag{4}$$

where $H = [h_1, h_2, \cdots, h_n] \in \mathbb{R}^{c\times m}$ is the label matrix of the data matrix Y and $h_i = [0, 0, \cdots, 1, \cdots, 0, 0]^T \in \mathbb{R}^c$ is the label vector of the sample Y_i, in which c is the number of classes and the position j of 1 in h_i is the class label of Y_i. The term $\|H - WA\|_F^2$ denotes the classification error [5], [30]. By solving this optimization problem, a recognition-driven dictionary D can be learnt.

B. Solving the Bilevel Optimization Problem

The stochastic subgradient descent algorithm [29] can be used to solve the optimization problem (4), but its convergence speed is relatively slow. Moreover, it is difficult to deduce the subgradient of the upper level objective function with respect to the dictionary D after implicitly representing W and A with D. In this paper, we use ADM [32] to solve it after some delicate reformulation.

We consider the lower level optimization:

$$\min_A \frac{1}{2}\|Y - DA\|_F^2 + \alpha\|A\|_1 + \frac{\beta}{2}\text{tr}\left(ALA^T\right). \tag{5}$$

Let $A = B - C$, where $B \in \mathbb{R}^{k\times n}$ and $C \in \mathbb{R}^{k\times n}$ are two nonnegative matrices such that B takes all the positive entries in A and the remaining entries of B are set to 0, while C does the same for the negative entries in A (after negation). Then problem (5) can be transformed into the following problem:

$$\begin{aligned} \min_{B,C} \quad & \frac{1}{2}\|Y - D(B - C)\|_F^2 + \frac{\beta}{2}\text{tr}\left((B - C)L(B - C)^T\right) \\ & + \alpha e_k^T(B + C)e_n, \\ \text{s.t.} \quad & B \ge 0, \quad C \ge 0, \end{aligned} \tag{6}$$

where $e_k \in \mathbb{R}^{k\times 1}$ and $e_n \in \mathbb{R}^{n\times 1}$ are two all-one vectors. $B \ge 0$ denotes that all the elements in matrix B are nonnegative. $C \ge 0$ has the same meaning. It should be pointed out that these two problems are equivalent.

Let $Z = [B; C] \in \mathbb{R}^{2k\times n}$ and $P = [I, -I] \in \mathbb{R}^{k\times 2k}$, in which $I \in \mathbb{R}^{k\times k}$ is the identity matrix, then we have $A = B - C = PZ$. Problem (6) can be rewritten as follows:

$$\begin{aligned} \min_Z \quad & \frac{1}{2}\|Y - DPZ\|_F^2 + \alpha e_{2k}^T Z e_n + \frac{\beta}{2}\text{tr}\left(PZLZ^T P^T\right), \\ \text{s.t.} \quad & Z \ge 0, \end{aligned} \tag{7}$$

where $e_{2k} \in \mathbb{R}^{2k\times 1}$ is an all-one vector.

Problem (7) is a convex problem, since its objective function is a sum of three convex functions and hence convex, and its constraint is a convex set. On the other hand, for any convex optimization problem with differentiable objective and constraint functions, KKT conditions are not only the necessary condition but also the sufficient condition of optimal solution [38]. Thus, if Z^* is an optimal solution to problem (7), Z^* must meet the KKT conditions of (7). Conversely, if Z^* meets the KKT conditions, then it is an optimal solution. So we can replace problem (7) with its KKT conditions. Write down the Lagrangian function of problem (7):

$$\begin{aligned} \mathcal{L}_1(Z, M) = & \frac{1}{2}\|Y - DPZ\|_F^2 + \frac{\beta}{2}\text{tr}\left(PZLZ^T P^T\right) \\ & + \alpha e_{2k}^T Z e_n + \text{tr}\left(M^T Z\right), \end{aligned} \tag{8}$$

where $M \in \mathbb{R}^{2k\times n}$ is the Lagrange multiplier matrix and M satisfies the constraint $M \le 0$. Since the constructed L is a symmetric matrix (please refer to Eq. (30)), we can obtain the KKT conditions of problem (7) as follows:

$$\begin{cases} P^T D^T DPZ - P^T D^T Y + \alpha E + \beta P^T PZL + M = 0, \\ M \odot Z = 0, \ Z \ge 0, \ M \le 0, \end{cases} \tag{9}$$

where $E \in \mathbb{R}^{2k\times n}$ is an all-one matrix.

Then we can replace the lower level optimization (5) with its KKT conditions (9) and obtain the following equivalent model:

$$\begin{aligned} \min_{W,Z,M,D} \quad & \|H - WPZ\|_F^2 + \lambda\|W\|_F^2, \\ \text{s.t.} \quad & P^T D^T DPZ - P^T D^T Y + \alpha E + \beta P^T PZL + M = 0, \\ & M \odot Z = 0, \quad Z \ge 0, \ M \le 0, \\ & \|D_i\|_2^2 \le 1, \quad \forall i \in \{1, 2, \cdots, k\}. \end{aligned} \tag{10}$$

The above problem is a unilevel optimization, hence can be solved by ADM. Since ADM does not enforce the constraints in each iteration (the constraints are exactly fulfilled only when convergence), this could be interpreted as that we do not have to solve the lower level optimization exactly in each iteration. As we can see, by ADM each variable can be updated with a closed form solution, rather than iteratively solving the lower level problem at a reasonably high precision as the (sub)gradient descent method does. Moreover, it is unnecessary to deduce the subgradient of the upper level objective function with respect to the dictionary D. So our new method is both faster and much simpler than the subgradient descent method.

To apply ADM, we first introduce two auxiliary variables X and S to update variables easily. The optimization problem (10) can be rewritten as

$$\begin{aligned} \min_{W,Z,M,X,S,D} \quad & \|H - WPZ\|_F^2 + \lambda\|W\|_F^2, \\ \text{s.t.} \quad & P^T D^T DPZ - P^T D^T Y + \alpha E + \beta XL + M = 0, \\ & M \odot S = 0, \ P^T PZ - X = 0, \ Z - S = 0, \\ & S \ge 0, \ M \le 0, \\ & \|D_i\|_2^2 \le 1, \ \forall i \in \{1, 2, \cdots, k\}. \end{aligned} \tag{11}$$

The augmented Lagrangian function of problem (11) is:

$$\begin{aligned}&\mathcal{L}_2(W, Z, M, X, S, D, R_1, R_2, R_3, R_4, \mu)\\&= \|H - WPZ\|_F^2 + \lambda\|W\|_F^2\\&\quad + \left\langle R_1, P^T D^T DPZ - P^T D^T Y + \alpha E + \beta XL + M\right\rangle\\&\quad + \langle R_2, M \odot S\rangle + \left\langle R_3, P^T PZ - X\right\rangle + \langle R_4, Z - S\rangle\\&\quad + \frac{\mu}{2}\|P^T D^T DPZ - P^T D^T Y + \alpha E + \beta XL + M\|_F^2\\&\quad + \frac{\mu}{2}\left(\|M \odot S\|_F^2 + \|P^T PZ - X\|_F^2 + \|Z - S\|_F^2\right), \quad (12)\end{aligned}$$

where $\langle A, B\rangle = \mathrm{tr}\left(A^T B\right)$, $R_1 \sim R_4$ are Lagrange multipliers, and $\mu > 0$ is the penalty parameter.

ADM updates the variables W, Z, M, S, X, and D alternately in each iteration, by minimizing the augmented Lagrangian function $\mathcal{L}_2$ with other variables fixed. Firstly, we update the parameter matrix W of the linear classifier.

$$W = HZ^T P^T (PZZ^T P^T + \lambda I)^{-1}. \quad (13)$$

Then, we update Z, M, S, and X in turn. More specifically, the iteration goes as follows:

$$\begin{aligned}Z =& \left(2P^T W^T WP + 2\mu P^T D^T DD^T DP + 2\mu P^T P + x\mu I\right)^{-1}\\&\times \Big[2P^T WH - \mu P^T D^T DP(-P^T D^T Y + \alpha E + \beta XL\\&+ M + R_1/\mu) - P^T P(R_3 - \mu X) - R_4 + \mu S\Big]. \quad (14)\end{aligned}$$

$$\begin{aligned}M =& -\Theta((S \odot R_2/\mu + P^T D^T DPZ - P^T D^T Y + \alpha E\\&\times + \beta XL + R_1/\mu) \oslash (S \odot S + E)), \quad (15)\end{aligned}$$

where $\Theta(\cdot)$ is an operator that projects a matrix onto the nonnegative cone, which can be defined as follows:

$$\Theta(X_{ij}) = \begin{cases} X_{ij}, & \text{if } X_{ij} \geq 0;\\ 0, & \text{otherwise.}\end{cases} \quad (16)$$

As for X and S, actually we can view (X, S) as a large block of variables. We can update (X, S) by minimizing the augmented Lagrangian function $\mathcal{L}_2$, which naturally splits into subproblems for X and S, respectively, since X and S are independent on each other in this minimization problem. Accordingly, we update these two variables as follows:

$$\begin{aligned}X =& \Big[P^T PZ + R_3/\mu - \beta(P^T D^T DPZ - P^T D^T Y\\&+ \alpha E + M + R_1/\mu)L^T\Big]\left(\beta^2 LL^T + I\right)^{-1}, \quad (17)\end{aligned}$$

$$S = \Theta\left((Z + R_4/\mu - M \odot R_2/\mu) \oslash (M \odot M + E)\right). \quad (18)$$

Now we focus on solving for D. We need to solve the following problem:

$$D = \arg\min_{D \in \Omega} \psi(D). \quad (19)$$

where $\Omega = \{D \mid \|D_i\|_2^2 \leq 1, i = 1, \cdots, k\}$ and

$$\begin{aligned}\psi(D) =& \|P^T D^T DPZ - P^T D^T Y + \alpha E + \beta XL + M\\&+ R_1/\mu\|_F^2. \quad (20)\end{aligned}$$

Algorithm 1 Solving the Bilevel Model for Discriminative Dictionary Learning (BMDDL) via ADM

Input: The training data matrix Y, the label matrix H of Y, the Laplacian matrix L of Y, the parameters λ, α, and β, max_iter$= 80$, maximal number loops in Armijo is 10.

1: Initialize D^0, Z^0, S^0, X^0 and M^0. Set $R_1^0 = 0$, $R_2^0 = 0$, $R_3^0 = 0$, $R_4^0 = 0$, $\mu^0 = 1e-3$, $\mu_{max} = 1e+8$, $\rho = 1.3$, $\varepsilon_1 = 1e-4$, $\varepsilon_2 = 1e-5$, and $j = 0$.
2: **while** not convergence **do**
3: Fix Z^j, M^j, X^j, S^j, and D^j to update W^{j+1} by (13).
4: Fix W^{j+1}, M^j, X^j, S^j, and D^j to update Z^{j+1} by (14).
5: Fix $W^{j+1}, Z^{j+1}, X^j, S^j$, and D^j to update M^{j+1} by (15).
6: Fix $W^{j+1}, Z^{j+1}, M^{j+1}, S^j$, and D^j to update X^{j+1} by (17).
7: Fix $W^{j+1}, Z^{j+1}, M^{j+1}, X^{j+1}$, and D^j to update S^{j+1} by (18).
8: Fix $W^{j+1}, Z^{j+1}, M^{j+1}, X^{j+1}$, and S^{j+1} to update D^{j+1} by (21).
9: Update Lagrange multipliers:

$$\begin{aligned}R_1^{j+1} =& R_1^j + \mu(\hat{R} + \alpha E + \beta X^{j+1}L + M^{j+1}),\\R_2^{j+1} =& R_2^j + \mu(M^{j+1} \odot S^{j+1}),\\R_3^{j+1} =& R_3^j + \mu(P^T PZ^{j+1} - X^{j+1}),\\R_4^{j+1} =& R_4^j + \mu(Z^{j+1} - S^{j+1}),\end{aligned}$$

where $\hat{R} = P^T(D^{j+1})^T D^{j+1} PZ^{j+1} - P^T(D^{j+1})^T Y$.
10: Update μ^{j+1}: $\mu^{j+1} = \min(\rho\mu^j, \mu_{max})$.
11: $j \leftarrow j + 1$.
12: Check convergence: if $\|P^T(D^j)^T D^j PZ^j - P^T(D^j)^T Y + \alpha E + \beta X^j L + M^j\|/\|E\| \leq \varepsilon_1$ and $\|M^j \odot S^j\|/\|M^j\| \leq \varepsilon_1$ and $\|P^T PZ^j - X^j\|/\|X^j\| \leq \varepsilon_1$ and $\|Z^j - S^j\|/\|Z^j\| \leq \varepsilon_1$ and $\|Z^j - Z^{j-1}\|/\|Z^j\| \leq \varepsilon_2$, then stop.
13: if $j \geq$max_iter, then stop.
14: **end while**

Output: The learnt dictionary D^j.

The problem (19) is a quartic polynomial minimization problem. It is difficult to compute its exact solution. So we use the projected gradient descent method [39] to update D:

$$D = \Pi_\Omega\left(\widehat{D} - \gamma\,\nabla\psi(\widehat{D})\right), \quad (21)$$

where $\widehat{D}$ is the previously computed value of D, the step size γ is chosen by the Armijo rule [40], and Π_Ω is the projection onto Ω.

The detailed optimization procedure of BMDDL is presented in **Algorithm 1**. The detailed deductions of the updates of W, Z, M, X, S, and D can be found in Supplementary Material.

C. Classification

When (11) is solved, we obtain a recognition-driven dictionary D and sparse codes $A = PZ$ of training samples.

Given a testing sample y, we first compute its sparse representation:

$$a^* = \arg\min_a \frac{1}{2}\|y - Da\|_F^2 + \alpha\|a\|_1 + \frac{\beta}{2}\sum_{i\in N_s(y)} q_i\|a - A_i\|_2^2, \tag{22}$$

where $N_s(y)$ denotes the set of s nearest neighbors of y. Note that the s nearest neighbors are chosen from training samples Y. q_i is the weight between training sample Y_i and y. A_i is the sparse code of the ith sample Y_i. α and β are regularization parameters. Actually, problem (22) is the vector form of the lower level problem (5). The values of α and β in problem (22) are the same as those in model (5), respectively. In this way, the sparse code learning problems in the training and the testing phases are consistent.

Problem (22) can be further written as follows:

$$a^* = \arg\min_a \frac{1}{2}\|OV_D^T a - O^{-1}V_D^T\tilde{y}\|_F^2 + \alpha\|a\|_1, \tag{23}$$

where $O = (\Sigma_D^T\Sigma_D + \beta\sum_{i\in N_s(y)} q_i I)^{\frac{1}{2}}$, $\tilde{y} = D^T y + \beta\sum_{i\in N_s(y)} q_i A_i$, and $U_D\Sigma_D V_D^T$ is the full singular value decomposition (SVD) of D. Therefore, we can apply any algorithm solving a LASSO problem, such as [12], [41], and [42], to solve (23). As LC-KSVD [5] and TDDL [30] did, we use the LARS [41] algorithm to solve (23) in this paper. Finally, we simply use the learnt linear classifier to estimate the label of a^*:

$$j^* = \arg\max_j \ (W^* a^*)_j, \tag{24}$$

where W^* is the parameter matrix of the learnt linear classifier.

D. Initialization

In Algorithm 1, we need to initialize D^0, Z^0, S^0, X^0, and M^0 first. Following LC-KSVD [5] and TDDL [30], we use several iterations of KSVD to learn a dictionary for each class and combine these small dictionaries together to form a dictionary D^0.

However, initializing Z^0 needs a little more effort. We initialize A^0 by solving problem (5), then we can compute $Z^0 = P^\dagger A^0$. We also adopt the ADM method [32] to solve (5). Firstly, we introduce two auxiliary variables J and G in order to update variables easily:

$$\begin{aligned} \min_{A,J,G} \quad & \frac{1}{2}\|Y - DJ\|_F^2 + \alpha\|A\|_1 + \frac{\beta}{2}\mathrm{tr}(GLG^T), \\ \text{s.t.} \quad & A = J, \quad A = G. \end{aligned} \tag{25}$$

Then the augmented Lagrangian function of problem (25) can be formulated as follows:

$$\begin{aligned} \mathcal{L}_3(A, J, G, R_5, R_6) = {} & \frac{1}{2}\|Y - DJ\|_F^2 + \alpha\|A\|_1 + \frac{\beta}{2}\mathrm{tr}(GLG^T) \\ & + \langle A - J, R_5\rangle + \frac{\mu}{2}\|A - J\|_F^2 \\ & + \langle A - G, R_6\rangle + \frac{\mu}{2}\|A - G\|_F^2, \end{aligned} \tag{26}$$

where R_5 and R_6 are Lagrange multipliers.

Algorithm 2 Solving Problem (5) for Initializing Z^0 via ADM

Input: The training data matrix Y, the Laplacian matrix L of Y, the parameters $\alpha > 0$ and $\beta > 0$, max_iter= 150.

1: Set $A^0 = J^0 = G^0 = D^\dagger Y$, $\mu^0 = 1e-2$, $\mu_{max} = 1e+8$, $\rho = 1.2$, $R_5^0 = R_6^0 = 0$, $\varepsilon = 1e-6$, and $j = 0$.
2: **while** not convergence **do**
3: Fix J^j and G^j to update A^{j+1} by (27).
4: Fix A^{j+1} and G^j to update J^{j+1} by (28).
5: Fix A^{j+1} and J^{j+1} to update G^{j+1} by (29).
6: Update Lagrange multipliers: $R_5 = R_5 + \mu^j(A^{j+1} - J^{j+1})$, $R_6 = R_6 + \mu^j(A^{j+1} - G^{j+1})$.
7: Update μ^{j+1}: $\mu^{j+1} = \min(\rho\mu^j, \mu_{max})$.
8: $j \leftarrow j + 1$.
9: Check convergence: if $\|A^j - J^j\|/\|A^j\| \le \epsilon$, $\|A^j - G^j\|/\|A^j\| \le \epsilon$ and $\|A^j - A^j\|/\|A^j\| \le \varepsilon$, then stop.
10: if $j \ge$max_iter, then stop.
11: **end while**

Output: $Z^0 = P^\dagger A^j$.

We update A, J, and G in turn. Note that we can also view (J, G) as a large block of variables since J and G are independent on each other in this minimization problem. Accordingly, we can update these three variables in the following way:

$$A = \mathcal{S}_{\alpha/(\mu)}\left(\frac{1}{2}(J + G - (R_5 + R_6)/\mu)\right), \tag{27}$$

where $\mathcal{S}_\epsilon(x) = \mathrm{sgn}(x)\max(|x| - \epsilon, 0)$ is the hard thresholding operator [43], and

$$J = V_D\left(\Sigma_D^T\Sigma_D + \mu I\right)^{-1} V_D^T\left(D^T Y + \mu A + R_5\right), \tag{28}$$

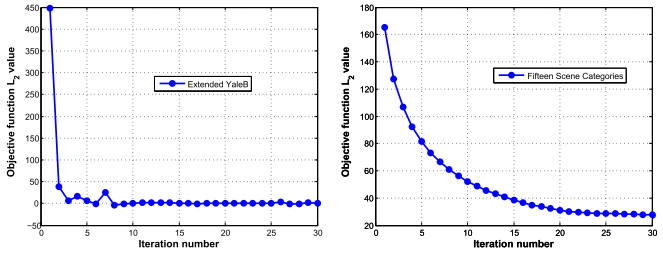
$$G = (\mu A + R_6)V_L\left(\Sigma_L + \mu I\right)^{-1} V_L^T, \tag{29}$$

where $U_D\Sigma_D V_D^T$ and $V_L\Sigma_L V_L^T$ are the full SVD of D and $\beta(L + L^T)/2$, respectively.

The procedure for solving problem (5) is described in **Algorithm** 2. The detailed deductions of the updates of A, J, and G can be found in Supplementary Material. After initializing Z^0, we can initialize $S^0 = Z^0$, $X^0 = P^T PZ^0$, and $M^0 = P^T(D^0)^T Y - P^T(D^0)^T D^0 PZ^0 - \alpha E - \beta X^0 L$.

E. Convergence Analysis

There is no theoretical convergence support when we apply ADM to solve problem (11). Typically, ADM for less than three blocks of variables usually converges when the problem is convex. Recently, some scholars propose theories to extend the scope of the convergence of ADM. For example, Hong and Luo [44] point out that ADM with K ($K \ge 3$) blocks of variables can converge when minimizing the sum of two or more nonsmooth convex separable functions which are subject to linear constraints. Hong et al. [45] also prove that ADM is convergent for a family of sharing problems, regardless of the number of blocks or the convexity of the objective function. Those works have extended the scope of ADM with theoretical guarantee. However, as for more complex optimization problems, which contain nonlinear equality constraints, are nonconvex and have K ($K \ge 3$) blocks of



(a) (b)

Fig. 1. Two examples of BMDDL minimization process, on Extended YaleB and Fifteen Scene Categories, respectively.

variables, there is no theory that supports the convergence of ADM. But this does not mean that ADM cannot converge. Boyd et al. [46] point out that when solving nonconvex problems by ADM, ADM may not converge, but when it does converge, it will possibly have better convergence properties than other local optimization methods. On the other hand, many scholars have also adopted ADM to solve nonconvex problems with nonlinear equality constraints and more than three blocks of variables, and they report state-of-the-art experimental results, such as [7]. To illustrate the convergence of ADM in solving problem (11), we conduct experiments and report in Fig. 1 (a) and (b) the objective value $\mathcal{L}_2$ on Extended YaleB [47] and Fifteen Scene Categories [48], respectively. We can see that the objective values reduce reasonably well.

IV. Unilevel, Bilevel and Multi-level

In this section, we first discuss in more detail the advantages of bilevel models over unilevel ones, then we generalize to multi-level models.

As we have mentioned in Section II-B, most supervised methods directly incorporate discriminative term $F(D, A, S)$ into the objective functions of unsupervised methods and the general supervised dictionary learning model can be formulated as (2). Such a mechanism leads to two drawbacks.

1) Undoubtedly, in recognition tasks, the classification error is our ultimate goal and we need to minimize it directly. However, these unilevel model based supervised methods [4], [5], [8], [18]–[23] minimize combinations of the reconstruction error and the discriminative terms, such as the classification error. In this way, the learnt dictionary is an optimal dictionary to the combined terms, rather than the classification error. Accordingly, the performance on recognition tasks may be compromised. On the contrary, bilevel models can overcome this drawback as they directly minimize the classification error. The upper level minimizes the classification loss, while the lower level characterizes the intrinsic data structure. The objective of lower level is subordinate to that of the upper level. Therefore, bilevel models achieve an overall optimality in that the dictionary learning is directly connected to recognition.

2) Another drawback of those unilevel model based methods [4], [5], [18], [20], [21], [23] is that the problems for computing the sparse codes in the training and the testing phases are different, making the models inconsistent. These methods for recognition tasks can be sketched in three steps. Firstly, these supervised methods solve problem (2) to learn a dictionary D, the sparse codes A_{tr} of the training samples, and other variables S, such as the classifier parameters in [4], [5], and [18]. Then, in the testing phase, since there is no supervision information, those methods have to discard the discriminative term $F(D, A, S)$ in (2) and fix dictionary D to compute the sparse codes A_{ts} of testing samples. Finally, these methods feed the feature A_{tr} of training samples into a classifier to learn its parameters W, then use W to identify the feature A_{ts} of testing samples. Or in [4], [5], and [18], they directly use the previously learnt classifier S of (2) in the training phase to classify testing samples. These methods solve different problems to learn the sparse representations A_{tr} of training samples and the sparse representations A_{ts} of testing samples. By this way, the new feature A_{ts} may not be optimal for the classifier W or S which is learnt on the feature A_{tr} of training samples. In contrast, bilevel models do not have the above problem. In the training phase, they solve the lower level optimization problem to compute the sparse representations A_{tr} of training samples, and in the testing phase, they still use the lower level model to compute the feature A_{ts} of testing samples. Thus, the classifier trained on the feature A_{tr} can perform on the feature A_{ts} of testing samples. So, in bilevel models the problems for computing the sparse codes in the training and the testing phases are consistent.

One could easily think of models with multiple levels. Then there are connections between bilevel models and supervised neural networks [49]–[51]. Both bilevel models and supervised neural networks are multi-level recognition-driven feature learning schemes. In recognition tasks, they both adopt the classification loss as their optimization goal and at each level, they both use a feature extractor, such as the lower level problem (5) in BMDDL, to learn discriminative features and feed them into the next level as input. But the feature extractors used in bilevel models are much more complex than those (linear mappings and nonlinear mappings) in neural networks, so that there are no closed-form solutions for the feature extractors. Please refer to Supplementary Material for further details.

V. Experiments

In this section, we evaluate our method on four different types of databases: Extended YaleB [47] (for face recognition), Fifteen Scene Categories [48] (for scene classification), Caltech 101 database [52] and Caltech 256 database [53] (for object recognition), and UCF50 [54] and HMDB51 [55] (for action recognition). As for the three parameters λ, α, and β in BMDDL, we select λ from the set $\{0.0001, 0.001, \cdots, 1\}$ and choose α and β from the sets $\{0.001, 0.004, 0.008, 0.01, \cdots, 1\}$ and $\{0.0001, 0.0005, 0.001, \cdots, 0.1\}$, respectively, in all experiments. Following [5], the parameters in our model are fixed for each database and determined by n-fold cross validation on the training data. The detailed parameter settings are presented in each experimental section. In the training

phase, we construct the weight matrix Q as follows:

$$Q_{ij} = \begin{cases} 1, \text{if samples } Y_i \text{ and } Y_j \text{ belong to the same class,} \\ 0, \text{otherwise.} \end{cases} \tag{30}$$

Then we compute its corresponding Laplacian matrix $L = T - Q$, where T is a diagonal matrix and $T_{ii} = \sum_j Q_{ij}$. In the testing phase, we find s nearest neighbors from training set for a testing sample. We set $s = 5$ and the weight $q_i = 1$ $(\forall i \in N_s(\cdot))$ in all experiments.

In all the above recognition tasks, we compare our method with supervised dictionary learning methods, including D-KSVD [4], LRRDL [7], SLRRDL [7], TDDL [30], LC-KSVD [5], DDL-PC [18], SRC [3], and unsupervised methods, such as KSVD [1], LSC [17], and SDL [14]. In each specific task, we further compare with other state-of-the-art methods with similar framework for that task, such as the classic locality-constrained linear coding (LLC) method [35]. The platform is Matlab 2013a under Windows 8 on a PC equipped with a 3.4GHz CPU and 16GB memory. **Our code will be released.**

A. Face Recognition

In this subsection, we conduct face recognition experiments on the widely used Extended YaleB [47]. It consists of 2,414 cropped frontal face images of 38 people. Every image has $192 \times 168 = 32,256$ pixels. There are between 59 and 64 images for each person. Following [5], we randomly select half of the samples of each person for training and the other half for testing. Since the dimension of the image feature is too high, each image feature is projected onto a 504-dimensional vector with a randomly generated matrix [5]. We take the dimension-reduced feature to evaluate D-KSVD [4], LRRDL [7], SLRRDL [7], TDDL [30], LC-KSVD1 [5], LC-KSVD2 [5], SRC [3], KSVD [1], SDL [14] and our method. Note that both LSC [17] and LLC [35] use SIFT descriptors [57], so we downsample these images by 4 such that the downsampled images can still produce a certain amount of SIFT features. LSC and LLC are the original LSC and LLC, respectively, while LSC* and LLC* use Laplacian sparse coding and sparse coding to encode the dimension-reduced feature, respectively. The dictionary size is 570. We set $\lambda = 0.001$, $\alpha = 1$, and $\beta = 0.005$ in our method. Every experiment runs 10 times and we report its average recognition rate.

The experimental results are summarized in Table III. We can see that our method obtains the best recognition rate by 0.5% more than the runner-up. We also note that most supervised methods achieve better classification performance than unsupervised methods, since supervised methods exploit the class discriminative information to learn a more discriminative dictionary for a specific task. Two of the discriminative sparse codes extracted from the Extended YaleB are shown in Fig. 2. We can see that the samples from the same class share a few atoms in the dictionary to linearly approximate themselves, which makes these features much easier to be identified.

TABLE III

THE RECOGNITION RATES (%) ON THE EXTENDED YALEB DATABASE ("SUP." AND "UNSUP." ARE SHORT FOR "SUPERVISED" AND "UNSUPERVISED", RESPECTIVELY)

Type	Method	Accuracy	Type	Method	Accuracy
Sup.	BMDDL (ours)	**95.5**	Unsup.	KSVD [1]	93.1
	D-KSVD [4]	94.1		LSC [17]	93.6
	LRRDL [7]	91.3		LSC* [17]	93.2
	SLRRDL [7]	91.9		SDL [14]	94.2
	TDDL [30]	94.6	Others	LLC [35]	82.2
	LC-KSVD1 [5]	94.5		LLC* [35]	88.5
	LC-KSVD2 [5]	95.0		Xu [56]	94.3
	SRC [3]	92.2			

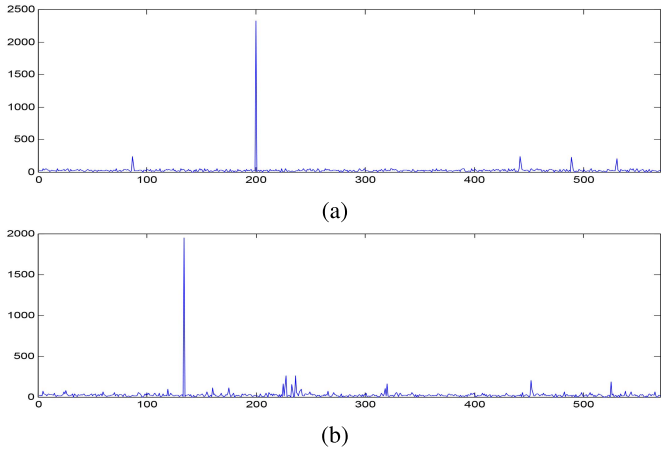


Fig. 2. Examples of sparse codes extracted from the Extended YaleB database. Each waveform denotes a sum of absolute sparse codes for different samples from the same class. Figures (a) and (b) correspond to two different classes.

Fig. 3. Examples of the Fifteen Scene Categories database.

B. Scene Classification

We use the Fifteen Scene Categories database [48] for scene classification. As shown in Fig. 3, this database contains a wide range of outdoor and indoor scenes, including office, kitchen, street, and coast. The size of each image is roughly 250×300 pixels. Each category contains about 200 to 400 images.

When we evaluate our method and other related methods on this database, we use the extracted features provided by [5]. The features are computed as follows. Firstly, we extract four-level spatial pyramid features, then encode these features with a codebook of size 200. Since the feature dimension is too high, PCA is used to reduce the feature dimension to 3,000. As [5] and [48] did, we randomly select 100 samples per category as training data and use the remaining samples for testing. For fairness, D-KSVD [4], LRRDL [7], SLRRDL [7], TDDL [30], LC-KSVD [5], SRC [3], KSVD [1], LSC* [17], SDL [14] and our method all use the spatial pyramid features and dictionary size is set as 450. Note that in [34], Lobel et al. use two kinds of features, HOG and

TABLE IV

THE RECOGNITION RATES (%) ON THE FIFTEEN SCENE CATEGORIES DATABASE ("SUP." AND "UNSUP." ARE SHORT FOR "SUPERVISED" AND "UNSUPERVISED", RESPECTIVELY)

Type	Method	Accuracy	Type	Method	Accuracy
Sup.	BMDDL (ours)	**96.9**	Unsup.	SDL [14]	88.1
	D-KSVD [4]	89.1	Others	LLC [35]	79.4
	LRRDL [7]	90.1		LLC* [35]	89.2
	SLRRDL [7]	91.3		Lazebnik [48]	81.4
	TDDL [30]	92.1		Gemert [58]	76.7
	LC-KSVD1 [5]	90.4		Yang [16]	80.3
	LC-KSVD2 [5]	92.9		Lian [59]	86.4
	SRC [3]	91.8		Boureau [60]	84.3
	Lobel [34]	86.3		Yang [61]	92.9
Unsup.	KSVD [1]	86.7		Wei [62]	91.8
	LSC [17]	89.9		Song [63]	85.7
	LSC* [17]	90.3			

TABLE V

THE RECOGNITION RATES (%) ON THE CALTECH 101 DATABASE ("SUP." AND "UNSUP." ARE SHORT FOR "SUPERVISED" AND "UNSUPERVISED", RESPECTIVELY)

Type	Method	Accuracy	Type	Method	Accuracy
Sup.	BMDDL (ours)	**75.5**	Unsup.	SDL [14]	70.2
	D-KSVD [4]	71.2	Others	LLC [35]	64.8
	LRRDL [7]	70.1		LLC* [35]	70.8
	SLRRDL [7]	71.0		Lazebnik [48]	64.6
	TDDL [30]	71.5		Gemert [58]	64.2
	LC-KSVD1 [5]	71.6		Geusebroek [64]	64.1
	LC-KSVD2 [5]	72.0		Y. Ng [65]	72.6
	SRC [3]	69.3		Malik [66]	56.6
	Lobel [34]	75.4		Zhang [67]	73.5
Unsup.	KSVD [1]	69.9		Quan [68]	68.4
	LSC [17]	69.2		Zhou [69]	75.2
	LSC* [17]	70.8			

LBP. We set both the neighborhood size of LLC and LLC* as 30. We set $\lambda = 0.0001$, $\alpha = 0.001$, and $\beta = 0.0001$ in our method.

The detailed comparison results are summarized in Table IV. Our method outperforms all the competing dictionary learning methods and other state-of-the-art methods. Our method makes about 4.0% improvement over the runner-up. The confusion matrix of our method can be found in Supplementary Material. There is no class that are classified badly and the worst recognition rate is as high as 90.7%.

C. Object Recognition

In our experiments, Caltech 101 [52] and Caltech 256 [53] are used to evaluate our method for object recognition.

1) Caltech 101: This database contains 9,146 images in total and includes 101 object categories (such as airplane, camera, face, ant, and piano) and an additional background category for a total of 102 categories. The number of each object category is between 31 to 800. The size of each image is roughly 300×200 pixels.

Following the same settings as in [5] and [7], we test our method with spatial pyramid features. We can take the following measures to extract these features. Firstly, we extract SIFT descriptors of 16×16 over a grid with a spacing of 8 pixels. Then, with three kind of grids with size 1×1, 2×2, and 4×4, we extract three-level spatial pyramid features based on the computed SIFT features. Finally, we encode the three-level spatial pyramid features with a codebook of size $1,024$. Since the feature dimension is too high, we reduce the feature dimension to $1,500$ with PCA. Following the common setup, we randomly select 30 samples per category as training data and use the remaining samples for testing. The detailed comparison results are reported in Table V. D-KSVD [4], LRRDL [7], SLRRDL [7], TDDL [30], LC-KSVD [5], SRC [3], KSVD [1], LSC* [17], SDL [14] and our method all use the extracted spatial pyramid features. The dictionary size is 3,060. LLC and LLC* both have 30 neighborhoods. In BMDDL, we set $\lambda = 1$, $\alpha = 0.008$, and $\beta = 0.0001$.

As Table V shows, our method achieves the best performance and makes about 2.9% improvement over the second best except Lobel [34]. Note that in [34], Lobel et al. use two kinds of features, HOG and LBP, while BMDDL only use SIFT. BMDDL still outperforms Lobel [34]. It is worth noting that in BMDDL, there are a total of sixteen classes that achieve the 100% recognition rate.

TABLE VI

THE RECOGNITION RATES (%) ON THE CALTECH 256 DATABASE ("SUP." AND "UNSUP." ARE SHORT FOR "SUPERVISED" AND "UNSUPERVISED", RESPECTIVELY)

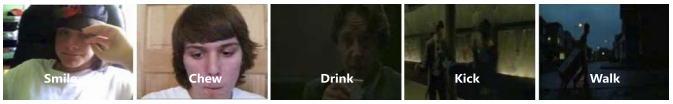
Type	Method	Accuracy	Type	Method	Accuracy
Sup.	BMDDL (ours)	**59.3**	Unsup.	LSC* [17]	57.5
	D-KSVD [4]	58.2		SDL [14]	57.8
	LRRDL [7]	57.4	Others	LLC [35]	41.9
	SLRRDL [7]	58.3		LLC* [35]	57.7
	TDDL [30]	57.6		Bo [70]	50.7
	LC-KSVD1 [5]	57.8		Liu [71]	45.7
	LC-KSVD2 [5]	58.6		Gao [17]	35.7
	SRC [3]	56.8		OverFeat [72]	56.4
Unsup.	KSVD [1]	57.3		Zhang [67]	46.3
	LSC [17]	34.9		Gao [73]	42.1

2) Caltech 256: This database consists of 30,607 images and splits between 256 distinct objects and a background category. Caltech 256 contains from 80 to 827 images per category. Compared with Caltech 101, it is more difficult due to its much higher intra-class and object location variability. Thus, we evaluate our method and other related methods with the OverFeat feature [72], which is 4,096 dimensional.

Following the common experimental settings, we randomly select 30 training images from each class and the remaining images are used for testing. For fairness, we evaluate D-KSVD [4], LRRDL [7], SLRRDL [7], TDDL [30], LC-KSVD [5], SRC [3], KSVD [1], LSC [17], SDL [14] and our method with OverFeat features and set the dictionary size as 3,855. LLC and LLC* have 30 and 15 neighborhoods, respectively. We set $\lambda = 1$, $\alpha = 0.001$, and $\beta = 0.001$ in our method.

The detailed comparison results are reported in Table VI, in which we compare our method with D-KSVD [4], LRRDL [7], SLRRDL [7], TDDL [30], LC-KSVD1 [5], LC-KSVD2 [5], SRC [3], KSVD [1], LSC [17], LLC [35], SDL [14] and other state-of-the-art object recognition approaches, [17], [70]–[72]. As can be seen from Table VI, our method outperforms the second best method by more than 0.7%.

(a)



(b)

Fig. 4. Samples of action recognition databases. (a) Samples of the UCF50 database. (b) Samples of the HMDB51 database.

TABLE VII

THE RECOGNITION RATES (%) ON THE UCF50 DATABASE ("SUP." AND "UNSUP." ARE SHORT FOR "SUPERVISED" AND "UNSUPERVISED", RESPECTIVELY)

Type	Method	Accuracy	Type	Method	Accuracy
Sup.	BMDDL (ours)	**73.2**	Unsup.	LSC* [17]	64.8
	D-KSVD [4]	65.9		SDL [14]	64.2
	LRRDL [7]	63.3	Others	LLC* [35]	60.9
	SLRRDL [7]	64.5		Gist [74]	38.8
	TDDL [30]	64.8		Laptev [75], [76]	47.9
	LC-KSVD1 [5]	64.9		Action Bank [77]	57.9
	LC-KSVD2 [5]	67.6		Zhang [78]	60.9
	SRC [3]	62.9		Liu [79]	62.7
	KSVD [1]	63.4			

D. Action Recognition

Finally, we test our method and other related methods for action recognition on the UCF50 database [54] and the HMDB51 database [55].

1) UCF50: The UCF50 database is one of the largest action recognition databases, consisting of realistic videos taken from YouTube. It contains 50 action categories with a total of 6,617 action videos and the categories are Baseball Pitch, Basketball Drumming, Biking, Diving, Tennis Swing, etc. Fig. 4 (a) shows some examples from this database.

For the UCF50 database, we use the action feature representations[1] [77] to evaluate our method and related methods. As the dimension of action feature is very high, we use PCA to reduce the feature dimension to 1,500. Then we take the dimension-reduced feature to evaluate our method, D-KSVD [4], LRRDL [7], SLRRDL [7], TDDL [30], LC-KSVD1 [5], LC-KSVD2 [5], SRC [3], KSVD [1], LSC [17], and SDL [14]. We follow the common experiment settings in [74]–[77] and test these methods with the five-fold group-wise cross-validation methodology. The dictionary size is 1,500. When we evaluate LSC* and LLC*, we use the original LSC and LLC methods to encode the action feature, respectively. The neighborhood number of LLC* is 30. In our method, we set $\lambda = 0.001$, $\alpha = 0.01$, and $\beta = 0.001$.

The detailed comparison results are summarized in Table VII. Our result is better than the competing dictionary learning methods and other state-of-the-art methods. Our method makes about 5.6% improvement over the runner-up.

[1]UCF50 feature: `http://www.cse.buffalo.edu/^jcorso/r/actionbank`.

TABLE VIII

THE RECOGNITION RATES (%) ON THE HDBM51 DATABASE ("SUP." AND "UNSUP." ARE SHORT FOR "SUPERVISED" AND "UNSUPERVISED", RESPECTIVELY)

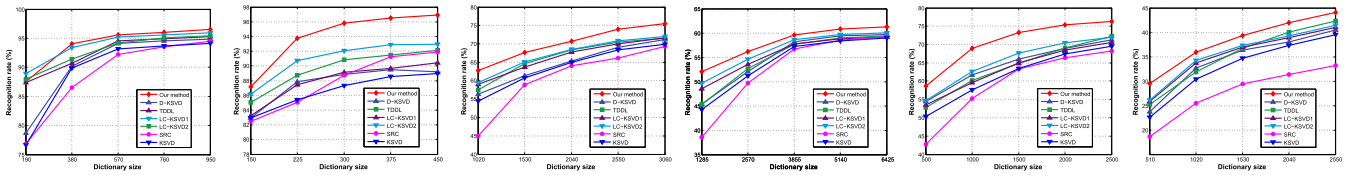
Type	Method	Accuracy	Type	Method	Accuracy
Sup.	BMDDL (ours)	**39.3**	Unsup.	SDL [14]	36.1
	D-KSVD [4]	36.4	Others	LLC* [35]	30.6
	LRRDL [7]	34.9		Kuehne [55]	20.0
	SLRRDL [7]	35.8		Action Bank [77]	26.9
	TDDL [30]	36.7		Kliper-Gross [80]	29.2
	LC-KSVD1 [5]	36.9		Solmaz [81]	29.2
	LC-KSVD2 [5]	37.3		Wang [82]	33.7
	SRC [3]	29.4		Zhang [83]	33.1
Unsup.	KSVD [1]	34.7		Sapienza [84]	37.2
	LSC* [17]	35.5			

2) HMDB51: The recently released HMDB51 is another large dataset for action recognition. It contains 6,849 clips divided into 51 action categories and each category contains a minimum of 101 clips. As shown in Fig. 4 (b), it consists of not only body movements but also facial actions, such as smile, laugh, chew, talk, and eat, which make it more difficult to be recognized.

We also employ the dimension-reduced action feature representations[2] [77] to evaluate D-KSVD [4], LRRDL [7], SLRRDL [7], TDDL [30], LC-KSVD1 [5], LC-KSVD2 [5], SRC [3], KSVD [1], LSC [17], SDL [14], and our method. The feature dimension is also 1,500. We follow the evaluation protocol of [55], [77], and [80]–[82], i.e., use three train/test splits, each with 70 training and 30 testing samples per class. The neighborhood number of LLC* is 30. The dictionary size is 1,530. We set $\lambda = 0.01$, $\alpha = 0.01$, and $\beta = 0.0005$ in our method. From Table VIII, we can see that our method obtains the best recognition rate by 2.0% more than the second best.

We also conduct experiments on the six testing databases to evaluate the performance of our method with different dictionary sizes. The experimental settings are as described in the above subsections, respectively. In the experiments, we evaluate our method, D-KSVD, TDDL, LC-KSVD1, LC-KSVD2, SRC, and KSVD. The experimental results are summarized in Fig. 5. We can see that with different dictionary sizes, our method consistently outperforms other six competing methods on all the six databases. These results clearly demonstrate that BMDDL is able to learn a more discriminative dictionary. Fig. 5 also demonstrates that supervised methods usually achieve better classification performance than unsupervised methods, especially when the dictionary size is small and the testing database is challenging. We also note that when the dictionary size reaches a certain scale, the recognition rate will not have a noticeable increase. However, the computing would be expensive when the dictionary size becomes large. Therefore, choosing an appropriate scale of dictionary is important for both achieving a good recognition performance and saving computation time. Note that TDDL [30] is also a bilevel model based dictionary learning method and it replaces the Laplacian term $\mathrm{tr}(ALA^T)$ in problem (4) with a regularization $\|A\|_F^2$. From Tables III∼VIII and Fig. 5, BMDDL achieves better

[2]HMDB51 feature: `http://www.cse.buffalo.edu/^jcorso/r/actionbank`.



(a) (b) (c) (d) (e) (f)

Fig. 5. Performance on the six testing databases with varying dictionary sizes. (a) Extended YaleB. (b) 15 Scene Categories. (c) Caltech 101. (d) Caltech 256. (e) UCF50. (f) HMDB51.

TABLE IX

THE AVERAGE TRAINING TIME (SECONDS) ON THE SIX DATABASES ("SUP." AND "UNSUP." ARE SHORT FOR "SUPERVISED" AND "UNSUPERVISED", RESPECTIVELY)

Type	Method	Extended YaleB	15 Scene Categories	Caltech 101	Caltech 256	UCF50	HMDB51
Sup.	BMDDL (ours)	**0.086**	0.152	2.274	2.862	0.267	0.335
	D-KSVD [4]	0.474	4.524	8.630	12.411	6.037	6.185
	LRRDL [7]	0.761	1.445	4.471	6.418	2.504	3.854
	SLRRDL [7]	0.809	1.665	5.101	7.328	2.974	3.689
	TDDL [30]	1.785	2.827	6.491	8.726	5.250	5.607
	LC-KSVD1 [5]	0.087	**0.145**	**1.711**	**2.563**	**0.243**	**0.321**
	LC-KSVD2 [5]	0.092	0.149	1.807	2.621	0.254	0.343
Unsup.	KSVD [1]	0.330	3.642	7.301	11.105	3.923	4.095
	SDL [14]	0.168	0.896	5.392	7.241	2.556	3.152

TABLE X

THE AVERAGE TESTING TIME (SECONDS) ON THE SIX DATABASES ("SUP." IS SHORT FOR "SUPERVISED")

Type	Method	Extended YaleB	15 Scene Categories	Caltech 101	Caltech 256	UCF50	HMDB51
Sup.	BMDDL (ours)	**0.021**	**0.044**	**0.438**	**0.521**	**0.254**	**0.275**
	LRRDL [7]	0.104	0.117	0.904	1.162	0.676	0.714
	SLRRDL [7]	0.107	0.123	0.933	1.235	0.727	0.753
	SRC [3]	0.110	0.137	7.102	10.059	3.521	3.437

performance than TDDL on the six benchmarks, which also demonstrates the advantages of the Laplacian regularization that encourages similar samples to have similar sparse codes.

E. Comparison of Computation Time

In the above subsections, we have compared our method with other state-of-the-art methods in terms of the recognition rate. In this subsection, we compare the average training and testing time of our method with those of D-KSVD [4], LRRDL [7], SLRRDL [7], TDDL [30], LC-KSVD1 [5], LC-KSVD2 [5], SRC [3], KSVD [1], and SDL [14] on the six testing databases. It should be pointed out that the experimental settings in this subsection are as described in the above subsections, respectively. The training time is defined as the time spent on training parameters of a model (it mainly contains the time for learning a dictionary). The testing time is the time from inputting a test sample to outputting its label. The average training time and testing time are computed as the training and testing time divided by the numbers of training samples and testing samples, respectively. Note that SRC has no training time and only has testing time, since it only needs to represent a testing sample as a linear combination of dictionary atoms, then uses the representation coefficients for recognition.

Table IX reports the average training time of these methods. LC-KSVD1, LC-KSVD2, and our method are the three fastest approaches. These three methods are about four times faster than the fourth fastest method on the Extended YaleB and roughly two times faster than the fastest of other methods on Caltech 101 and Caltech 256. They are also at least more than ten times faster than other compared methods on the remaining three databases. Note that all the methods cost much more training time on Caltech 101 and Caltech 256 than other databases. The reason is that the size of these two databases is large and it will be much more computationally expensive for each iteration. Note that TDDL [30] replaces the Laplacian term $\mathrm{tr}(ALA^T)$ in problem (4) with a regularization $\|A\|_F^2$, which results in a subproblem that is easier to solve for the subgradient with respect to D via implicit differentiation. From Table IX, we can see that though our optimization method solves a more complex problem, our method is still faster than TDDL, which demonstrates that our optimization method runs faster than the stochastic subgradient descent algorithm.

The average testing time on the six databases are summarized in Table X. The testing phases of D-KSVD, LC-KSVD1, LC-KSVD2, TDDL, KSVD and SDL are similar to ours. Namely, these methods and our method all need to solve a LASSO problem when they compute the sparse representation of a testing sample. Since the testing databases are not very large, compared with the time for solving a LASSO problem, the time for computing k nearest neighbors in our method can be negligible. So we only select LRRDL, SLRRDL, and SRC as our competitors. Table X shows that our method is the

TABLE XI

THE COMPARISON OF RECOGNITION RATES (%) BETWEEN UNILEVEL AND BILEVEL ON FOUR DATABASES ("UNI." AND "BI." ARE SHORT FOR "UNILEVEL" AND "BILEVEL", RESPECTIVELY. "YALEB" AND "15 SCENE" DENOTE "EXTENDED YALEB" AND "15 SCENE CATEGORIES", RESPECTIVELY)

Type	Method	YaleB	15 Scene	Caltech 101	Caltech 256
Uni.	DDL-PC [18]	95.3	92.0	71.3	58.3
Uni.	UMDDL (ours)	95.2	93.3	72.4	58.7
Bi.	BMDDL (ours)	**95.5**	**96.9**	**75.5**	**59.3**

fastest. It is about two times faster than the second fastest method, LRRDL, on the six testing databases. It is more than two times faster than SRC on the Extended YaleB and 15 Scene Categories database and at least ten times faster than SRC on the remaining four databases.

F. Comparison Between Unilevel and Bilevel

To verify the advantages of bilevel model based method, we compare BMDDL with DDL-PC [18], and our unilevel model UMDDL, whose objective function is a combination of those of BMDDL. As previously mentioned, DDL-PC also incorporates a Laplacian term with a linear classifier to jointly learn a discriminative dictionary and a classifier. However, DDL-PC is unilevel. Its training model can be found in Table II in Section II-B, and its testing model is problem (22) after discarding the Laplacian term, i.e., setting $\beta = 0$. The training model of UMDDL is the same as DDL-PC, while the testing model is problem (22). DDL-PC uses the feature-sign search algorithm [12] to optimize its model, while UMDDL employs ADM. We summarize their experimental results on Extended YaleB, 15 Scene Categories, Caltech 101, and Caltech 256 in Table XI. The experimental settings in this subsection are as described in the above subsections, respectively. We can see that BMDDL outperforms both DDL-PC and UMDDL, since as we mentioned in Section IV, our bilevel model, BMDDL, directly minimizes the classification loss and its models for computing sparse codes in the training and the testing phases are consistent.

VI. CONCLUSIONS AND FUTURE WORK

We propose a novel bilevel model based discriminative dictionary learning method for recognition tasks. Unlike other supervised dictionary learning methods that optimize the combination of the reconstruction error, the sparsity of representation, and other discriminative terms, our method directly minimizes the classification error at the upper level. The lower level optimizes the reconstruction error and group sparsity of representation. The lower level is subordinate to the upper one, rather than in parallel. In this way, the learnt dictionary may be optimal for recognition and it preserves the structure information of data at the same time. Moreover, the problems for computing the sparse codes in the training and the testing phases can be the same, making a consistent learning model. Finally, we propose a novel method to solve our bilevel optimization problem. We utilize the KKT conditions and ADM to reformulate our model and solve the equivalent model, respectively. Extensive experimental results demonstrate that our method obtains better classification results than other dictionary learning methods and some task-specific recognition methods, even with a simple linear classifier.

In the future, in the same spirit we will use more sophisticated classification losses, instead of the ridge regression classification error, at the upper level to learn more discriminative dictionaries for recognition tasks. We will also explore the convergence issue of ADM when applying it to nonconvex optimization problems that have nonlinear linear constraints and $K \geq 3$ blocks of variables.

REFERENCES

[1] 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.

[2] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[3] 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.

[4] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2691–2698.

[5] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[6] C.-F. Chen, C.-P. Wei, and Y.-C. F. Wang, "Low-rank matrix recovery with structural incoherence for robust face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2618–2625.

[7] Y. Zhang, Z. Jiang, and L. S. Davis, "Learning structured low-rank representations for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 676–683.

[8] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 209–232, Sep. 2014.

[9] Y. Sun, Q. Liu, J. Tang, and D. Tao, "Learning discriminative dictionary for group sparse representation," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3816–3828, Sep. 2014.

[10] K. Engan, S. O. Aase, and J. H. Husoy, "Frame based signal compression using method of optimal directions (MOD)," in *Proc. IEEE Int. Symp. Circuits Syst.*, May/Jun. 1999, pp. 1–4.

[11] 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.

[12] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Proc. Conf. Neutral Inf. Process. Syst.*, 2006, pp. 801–808.

[13] R. Jenatton, J. Mairal, F. Bach, and G. Obozinski, "Proximal methods for sparse hierarchical dictionary learning," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 1–8.

[14] A.-K. Seghouane and M. Hanif, "A sequential dictionary learning algorithm with enforced sparsity," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2015, pp. 3876–3880.

[15] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 759–766.

[16] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1794–1801.

[17] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 92–104, Jan. 2013.

[18] H. Guo, Z. Jiang, and L. S. Davis, "Discriminative dictionary learning with pairwise constraints," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 328–342.

[19] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Proc. Conf. Neutral Inf. Process. Syst.*, 2009, pp. 1033–1040.

[20] X.-C. Lian, Z. Li, C. Wang, B.-L. Lu, and L. Zhang, "Probabilistic models for supervised dictionary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2305–2312.

[21] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3517–3524.

[22] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, "Max-margin multiple-instance dictionary learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 846–854.

[23] N. Zhou, Y. Shen, J. Peng, and J. Fan, "Learning inter-related visual dictionary for object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3490–3497.

[24] J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing," *Found. Trends Comput. Graph. Vis.*, vol. 8, nos. 2–3, pp. 85–283, 2014.

[25] M. Kowalski, "Sparse regression using mixed norms," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 303–324, 2009.

[26] E. Elhamifar and R. Vidal, "Robust classification using structured sparse representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1873–1879.

[27] J. Huang and T. Zhang, "The benefit of group sparsity," *Ann. Statist.*, vol. 38, no. 4, pp. 1978–2004, 2010.

[28] F. Wang, N. Lee, J. Sun, J. Hu, and S. Ebadollahi, "Automatic group sparse coding," in *Proc. Int. Conf. Learn. Represent.*, 2011, pp. 495–500.

[29] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer, 1997.

[30] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[31] J. Yang, Z. Wang, Z. Lin, X. Shu, and T. Huang, "Bilevel sparse coding for coupled feature spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2360–2367.

[32] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Conf. Neutral Inf. Process. Syst.*, 2011, pp. 612–620.

[33] L. Tao, F. Porikli, and R. Vidal, "Sparse dictionaries for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 549–564.

[34] H. Lobel, R. Vidal, and A. Soto, "Learning shared, discriminative, and compact representations for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 11, pp. 2218–2231, Nov. 2015.

[35] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained Linear Coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.

[36] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915–936, 2003.

[37] G. H. Golub, P. Hansen, and D. O'Leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 1, pp. 185–194, 1999.

[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[39] G. P. McCormick and R. A. Tapia, "The gradient projection method under mild differentiability conditions," *SIAM J. Control*, vol. 10, no. 1, pp. 93–98, 1972.

[40] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific J. Math.*, vol. 16, no. 1, pp. 1–3, 1966.

[41] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[42] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[43] S. Foucart, "Hard thresholding pursuit: An algorithm for compressive sensing," *SIAM J. Numer. Anal.*, vol. 49, no. 6, pp. 2543–2563, 2011.

[44] M. Hong and Z.-Q. Luo. (2012). "On the linear convergence of the alternating direction method of multipliers." [Online]. Available: https://arxiv.org/abs/1208.3922

[45] M. Hong, Z. Luo, and M. Razaviyayn, "Convergence analysis of ADMM for a family of nonconvex problems," in *Proc. Conf. Neutral Inf. Process. Syst.*, 2014, pp. 1–29.

[46] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[47] 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.

[48] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2169–2178.

[49] Y. LeCun, L. Bottou, G. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*. Springer, 1998, pp. 9–48.

[50] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[51] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[52] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, p. 178.

[53] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007.

[54] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of Web videos," *Mach. Vis. Appl.*, vol. 24, no. 5, pp. 971–981, 2013.

[55] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2556–2563.

[56] Y. Xu, Z. Zhong, J. Yang, J. You, and D. Zhang, "A new discriminative sparse representation method for robust face recognition via ℓ_2 regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, no. 99, pp 1–10, Jun. 2016.

[57] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[58] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 696–709.

[59] X.-C. Lian, Z. Li, B.-L. Lu, and L. Zhang, "Max-margin dictionary learning for multiclass image categorization," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 157–170.

[60] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2559–2566.

[61] S. Yang and D. Ramanan, "Multi-scale recognition with DAG-CNNs," in *Proc. IEEE Int. Conf. Comput. Vis.*, May 2015, pp. 1215–1223.

[62] X.-S. Wei, B.-B. Gao, and J. Wu, "Deep spatial pyramid ensemble for cultural event recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 280–286.

[63] X. Song, S. Jiang, and L. Herranz, "Joint multi-feature spatial context for scene recognition on the semantic manifold," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1312–1320.

[64] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1271–1283, Jul. 2010.

[65] C. Adam and A. Y. Andrew, "The importance of encoding versus training with sparse coding and vector quantization," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 921–928.

[66] S. Maji, A. C. Berg, and J. Malik, "Efficient classification for additive kernel SVMs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 66–77, Jan. 2013.

[67] C. Zhang, J. Cheng, J. Liu, J. Pang, Q. Huang, and Q. Tian, "Beyond explicit codebook generation: Visual representation using implicitly transferred codebooks," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5777–5788, Dec. 2015.

[68] Y. Quan, Y. Xu, Y. Sun, Y. Huang, and H. Ji, "Sparse coding for classification via discrimination ensemble," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2016, pp. 5839–5847.

[69] P. Zhou, Z. Lin, and C. Zhang, "Integrated low-rank-based discriminative feature learning for recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 1080–1093, May 2016.

[70] L. Bo, X. Ren, and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 660–667.

[71] B.-D. Liu, Y.-X. Wang, B. Shen, Y.-J. Zhang, and M. Hebert, "Self-explanatory sparse representation for image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 600–616.

[72] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2014.

[73] S. Gao, L. Duan, and I. W. Tsang, "DEFEATnet—A deep conventional image representation for image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 3, pp. 494–505, Mar. 2016.

[74] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[75] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[76] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 1–11.
[77] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1234–1241.
[78] B. Zhang, A. Perina, V. Murino, and A. D. Bue, "Sparse representation classification with manifold constraints transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4557–4565.
[79] Q. Liu and C. Liu, "A novel locally linear KNN model for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1329–1337.
[80] G. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 256–269.
[81] B. Solmaz, S. M. Assari, and M. Shah, "Classifying Web videos using a global video descriptor," *Mach. Vis. Appl.*, vol. 24, no. 7, pp. 1473–1485, Oct. 2013.
[82] L. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2674–2681.
[83] L. Zhang, Y. Feng, J. Han, and X. Zhen, "Realistic human action recognition: When deep learning meets VLAD," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2016, pp. 1352–1356.
[84] M. Sapienza, F. Cuzzolin, and P. H. S. Torr, "Learning discriminative space–time action parts from weakly labelled videos," *Int. J. Comput. Vis.*, vol. 110, no. 1, pp. 30–47, Oct. 2014.

Pan Zhou received the Bachelor of Computer Science and Technology degree from China University of Geosciences, Wuhan, China, in 2013. He is currently pursuing the master's degree with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. His research interests include computer vision, machine learning, and pattern recognition.

Chao Zhang received the Ph.D. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 1995. From 1995 to 1997, he was a Post-Doctoral Research Fellow with the National Laboratory on Machine Perception, Peking University. Since 1997, he has been an Associate Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. His research interests include image processing, statistical pattern recognition, and visual recognition.

Zhouchen Lin (M'00–SM'08) received the Ph.D. degree in applied mathematics from Peking University, in 2000. He was a Guest Professor with Shanghai Jiao Tong University, Beijing Jiaotong University, and with Southeast University. He was also a Guest Researcher with the Institute of Computing Technology, Chinese Academic of Sciences. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. He is also a Chair Professor with Northeast Normal University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an IAPR Fellow. He is an Area Chair of the CVPR 2014/2016, the ICCV 2015, the NIPS 2015, and a Senior Committee Member of the AAAI 2016/2017 and the IJCAI 2016. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*.