Neurocomputing 238 (2017) 103–113

Contents lists available at ScienceDirect

Neurocomputing

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

Synthesis linear classifier based analysis dictionary learning for pattern classification☆

Jiujun Wang, Yanqing Guo*, Jun Guo, Ming Li, Xiangwei Kong

School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, Liaoning Province, China

ARTICLE INFO

Article history:
Received 26 February 2016
Revised 13 January 2017
Accepted 16 January 2017
Available online 4 February 2017

Communicated by Prof. Y. Liu

Keywords:
Analysis dictionary learning
Synthesis linear classifier
Pattern classification

ABSTRACT

Dictionary learning approaches have been widely applied to solve pattern classification problems and have achieved promising performance. However, most of works aim to learn a discriminative synthesis dictionary and sparse coding coefficients for classification. Until recent years, analysis dictionary learning began to attract interest from researchers. In this paper, we present a novel discriminative analysis dictionary learning frame, named Synthesis Linear Classifier based Analysis Dictionary Learning (SLC-ADL). Firstly, we incorporate a synthesis-linear-classifier-based error term into the basic analysis dictionary learning model, whose classification performance is obviously improved by making full use of the label information. Then, we develop an alternating iterative algorithm to solve the new model and obtain closed-form solutions leading to pretty competitive running efficiency. What is more, we design three classification schemes by fully exploiting the synthesis linear classifier. Finally, extensive comparison experiments on scene categorization, object classification, action recognition and face recognition clearly verify the classification performance of the proposed algorithm.

1. Introduction

In recent years, sparse representation (SR) [1] has attracted much attention because of its successful applications in image categorization [2,3], visual saliency [4], image segmentation [5] and other computer vision problems [6–8]. SR represents a sample by the linear combination of several dictionary's atoms chosen out of the whole dataset (e.g., the classic sparse representation based classification [1]) or other predefined bases [9,10], so the dictionary is crucial in the sample reconstruction process. Learning an optimal dictionary from data instead of using a set of predefined bases has achieved a big success in various practical tasks, such as face recognition [11], image classification [12], image and speech denoising [13,14], and others [15].

According to the ways of encoding input data, dictionary learning (DL) approaches [16] can be mainly classified as synthesis dictionary learning (SDL) and analysis dictionary learning (ADL). The learned dictionaries are named synthesis dictionaries and analysis dictionaries, respectively.

SDL is the mainstream of DL research. It aims to learn a dictionary to synthesize an input sample by the linear combination of a few dictionary atoms. Most of the existing DL approaches belong to this category. K-SVD [9] is a representative method to learn an over-complete synthesis dictionary from natural image examples and leads to more promising performance in image reconstruction than the prespecified dictionaries. Based on K-SVD, Zhang et al. [11] proposed a discriminative K-SVD method (D-KSVD) to learn a desired dictionary, which retains the representation power of K-SVD while supporting optimal discrimination of the classes by directly incorporating the class labels in DL stage. Jiang et al. [17] further added a label consistency constraint to make sparse coding coefficients more discriminative, thus the learned dictionary was both reconstructive and discriminative. The above synthesis dictionaries learned from data are common shared dictionaries since data from different classes are represented by a single dictionary. Besides, another popular line in SDL aims to learn structured dictionaries to enhance discrimination between classes. The structured sub-dictionary is learned for each specific class and atoms out of the same sub-dictionary have the same class label. Following this line, Yang et al. [18] learned a class-specific dictionary for each class and achieved better face recognition performance than traditional sparse representation based classification (SRC) choosing data samples to build up the dictionary. To make the

☆ This work is funded by the National Natural Science Foundation of China (Grant nos. 61402079, 61172109), the Foundation for Innovative Research Groups of the NSFC (Grant no. 71421001), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (Grant no. 201600022), and the Fundamental Research Funds for the Central Universities (Grant no. DUT14RC(3)103).

* Corresponding author.
E-mail address: guoyq@dlut.edu.cn (Y. Guo).

http://dx.doi.org/10.1016/j.neucom.2017.01.041
0925-2312/

sub-dictionaries learned from different categories to be independent, Ramirez et al. [19] introduced dictionary learning with structured incoherence (DLSI). Furthermore, based on the structured dictionary, Yang et al. [20] imposed Fisher discrimination criterion on the coding coefficients to make within-class scatter small but between-class scatter large. Then, they used both the reconstruction error and discriminative SR coefficients for pattern classification. In order to integrate the advantages of shared dictionaries and structured dictionaries, the hybrid DL frameworks come into being. Kong and Wang [21] proposed a hybrid model named DL-COPAR, in which both class-specific dictionaries and shared pattern pool contribute to the improvement of discrimination. Zhou et al. [22] added a discriminative Fisher criterion on the representation coefficients to enhance the discrimination of the hybrid dictionary.

Although SDL has led to promising results, it cannot offer an intuitive explanation and it is time-consuming. Hence, analysis dictionaries, as the dual form of synthesis dictionaries, can offer a more intuitive illustration like feature transformation (e.g. DWT) for the role of dictionaries, begin to attract the attention from researchers.

Analysis dictionaries map training data to coding coefficients. The analysis K-SVD [23] frame aims to learn an analysis dictionary from a set of training examples and is a dual viewpoint of K-SVD algorithm that serves the corresponding problem in synthesis models. In addition, Ravishankar and Bresler [24] showed that well-conditioned square transformations are superior over the conventional transformations (e.g. DCT) for image representation and denoising. After that, Shekhar et al. [25] added a full-rank constraint to the analysis dictionary to enhance this method. Rubinstein and Elad [26] imposed a thresholding process on coding coefficients to constrain sparse coding coefficients for the reconstruction of input data. Gu et al. [27] integrated structured synthesis and analysis dictionaries together for pattern classification.

Most aforementioned DL methods aim to promote the discrimination power of synthesis and analysis dictionaries, or to reduce computational complexity and improve training and testing efficiency. The hierarchical dictionary tree based classifier is proposed [28] in recent years. Even so, the design of classifiers used in DL framework is still worth studying. As far as we know, the classifiers incorporated into DL are mainly the analysis classifiers [17]. The label vector of input data approximates the product of a desirable analysis classifier and the corresponding coding coefficient. To improve the discriminability of ADL model, we introduce a new synthesis linear classifier (SLC) to build up an error term, and then incorporate it into the basic ADL model. Thanks to SLC, the class information of the samples can be regarded as part of the feature. The label vectors are projected to the codes by SLC, which makes the codes more discriminative. Compared to the conventional classifier, SLC fully exploits the class information in a more intuitive way. Therefore, we can obtain a more discriminative and intuitive transform model. The new model can be solved by an alternating iterative method. Furthermore, we utilize three classification schemes based on the learned synthesis linear classifier to perform the classification tasks.

In this paper, our main contributions are as follows:

- We propose a novel synthesis linear classifier, which can be regarded as the dual form of a classical linear classifier, and then design three different classification schemes by fully exploiting the synthesis linear classifier.
- We present a novel discriminative ADL approach for pattern classification, named Synthesis Linear Classifier based Analysis Dictionary Learning (SLC-ADL). In our proposed model, a synthesis-linear-classifier-based error term is incorporated into the traditional ADL model to promote its discrimination power.
- We develop an alternating iterative algorithm to solve the new SLC-ADL model and obtain closed-form solutions, which can reduce the computational complexity. Consequently, we acquire pretty competitive running efficiency.
- Extensive experiments are conducted on commonly used databases, and the results validate that our proposed SLC-ADL method outperforms the state-of-the-art SDL methods in pattern classification tasks.

The organization of this paper is as follows. In Section 2, we briefly review SDL, the linear classifier and ADL. In Section 3, our proposed SLC-ADL model is introduced in detail. Section 4 gives three synthesis-linear-classifier-based classification schemes. The experimental settings and results are described in Section 5. In Section 6, we conclude this paper.

2. Related work

2.1. SDL and the linear classifier

The approaches of representing samples in a synthesis dictionary have attracted considerable attention recently. Given a training sample $\mathbf{y} \in \Re^n$, SDL hopes to learn an optimized synthesis dictionary $\mathbf{D}$. The $\mathbf{D}$ satisfies $\mathbf{y} \cong \mathbf{Dx}$, where $\mathbf{x}$ denotes the coding coefficient of $\mathbf{y}$ over the learned $\mathbf{D}$. Let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \Re^{n \times N}$ be data matrix with n-dimensional N input samples and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \Re^{m \times N}$ be corresponding sparse code matrix of $\mathbf{Y}$. Based on the conventional synthesis SR model, a synthesis reconstructive dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m] \in \Re^{n \times m}$ and the adaptive sparse code $\mathbf{X}$ can be obtained by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{D},\mathbf{X}} \quad & \|\mathbf{Y} - \mathbf{DX}\|_F^2 \\ s.t. \quad & \mathbf{D} \in \mathbf{\Lambda}, \\ & \|\mathbf{x}_i\|_0 \le T, \quad i = 1, 2, \ldots, N, \end{aligned} \tag{1}$$

where $\|\mathbf{Y} - \mathbf{DX}\|_F^2$ represents the reconstruction error and $\mathbf{\Lambda}$ is a constraint set about $\mathbf{D}$ to make solutions well-regularized. The parameter T is a positive integer constraining the sparsity of coding coefficients (the number of nonzero elements). The synthesis dictionary $\mathbf{D}$ is solved by minimizing the reconstruction error while meeting sparsity constraints. Usually, m, the number of the synthesis dictionary's atoms is larger than n, the dimensionality of input data, which means the dictionary is over-complete.

According to the above model, it is easy to find that the learned dictionary is only required to well reconstruct training samples, without contributing to the discriminative ability for a classification task. Efforts have been made to improve the DL procedure for classification tasks. Zhang and Li [11] proposed a discriminative K-SVD (D-KSVD) algorithm by adding a simple classifier into the model (1) in consideration of the classifier's performance. The formulation containing a classifier is

$$\min_{\mathbf{W}} \ \|\mathbf{H} - \mathbf{WX}\|_F^2 + \eta \|\mathbf{W}\|_F^2, \tag{2}$$

where $\mathbf{W}$ is a linear classifier and accomplishes $\mathbf{H} \cong \mathbf{WX}$. $\mathbf{H}$ is the label matrix of training data. The ith column vector of $\mathbf{H}$ is $\mathbf{h}_i = [0, 0, \ldots, 1, \ldots, 0, 0]^T$, where the location of the only one non-zero element indicates the class label. As a result, $\|\mathbf{H} - \mathbf{WX}\|_F^2$ can represent the classification error, and $\|\mathbf{W}\|_F^2$ is a regularization penalty term to prevent over-fitting.

Combining the formulation (1) and (2), D-KSVD [11] can be formulated as

$$\min_{\mathbf{D},\mathbf{W},\mathbf{X}} \ \|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2+\mu\|\mathbf{H}-\mathbf{W}\mathbf{X}\|_F^2+\nu\|\mathbf{W}\|_F^2$$
$$s.t. \quad \mathbf{D}\in\mathbf{\Lambda}, \qquad (3)$$
$$\|\mathbf{x}_i\|_0\le T, \quad i=1,2,\ldots,N,$$

where μ and ν are scalar parameters controlling the weight of corresponding terms.

To obtain discriminative sparse codes, Jiang et al. [17] developed D-KSVD to Label Consistent K-SVD (LC-KSVD) by adding a label-consistency term. In [17], they also used $\|\mathbf{H}-\mathbf{W}\mathbf{X}\|_F^2$ to represent the classification error. What's more, they proposed the pseudo-inverse method and joint optimization method respectively to obtain the linear classifier $\mathbf{W}$.

In pattern classification applications, to enhance the discriminative power of synthesis dictionaries, there are many various improved methods, such as DLSI [19], Fisher Discrimination Dictionary Learning (FDDL) [20], LC-KSVD [17]. Although SDL has achieved promising results, it cannot offer an intuitive explanation for coding and has a heavy computation burden.

2.2. ADL

ADL, as a dual viewpoint of the common SDL, offers a pretty intuitive understanding for coding like feature transformation and has a high speed to handle data. Given a training sample $\mathbf{y}\in\Re^n$, ADL aims to learn an analysis dictionary $\mathbf{\Omega}$. The $\mathbf{\Omega}$ minimizes $\|\mathbf{x}-\mathbf{\Omega}\mathbf{y}\|_F^2$ subjected to $\|\mathbf{x}\|_0\le T$, where $\mathbf{x}$ be sparse coding coefficient of $\mathbf{y}$. Inspired by [29] and [30], the sparsity constraint of l_0-norm can be achieved by a thresholding function operation.

Therefore, the analysis dictionary $\mathbf{\Omega}\in\Re^{m\times n}$ can be learned by solving

$$\min_{\mathbf{\Omega},\mathbf{X}} \ \|\mathbf{X}-\mathbf{\Omega}\mathbf{Y}\|_F^2$$
$$s.t. \quad \mathbf{\Omega}\in\Gamma, \qquad (4)$$
$$\|\mathbf{x}_i\|_0\le T, \quad i=1,2,\ldots,N.$$

To make sure solutions solvable and well-regularized, the set $\mathbf{\Gamma}$ can be constrained to unity row-wise norm or relatively small Frobenius norm, which is indicated by [25]. In [24,31], a generalized model, called transform learning, is proposed. The transform model assumes error living in the transform domain and suggests that the data can be sparse in the transform domain. Different from ADL, the sparse code in transform model is not constrained to lie in the range space of $\mathbf{\Omega}$ [24]. Because this model is more generalized, it works better for representation. What's more, it has led to further applications [32] and further efficient analysis operator learning algorithms [33]. Different from a synthesis dictionary, it is the number of row vectors in an analysis dictionary that is larger than the dimensionality of input data, which means $m>n$.

Although ADL has high testing efficiency since the coding coefficients can be obtained by matrix multiplication and a thresholding function operation, its discriminability is so poor that it cannot be applied to pattern classification tasks. Inspired by the improvement of synthesis dictionaries, we make full use of label information of data and incorporate a significant synthesis-linear-classifier-based error term into model (4) to promote its discriminability. Our frame will be introduced in detail in next section.

3. The proposed SLC-ADL model

Although ADL gives an intuitive meaning for coding and has a high speed to handle data, it still needs further improvement on its discriminative ability to accomplish pattern classification tasks. In this section, we introduce our proposed SLC-ADL model to enhance its discriminability, which utilizes the class information by adding a synthesis-linear-classifier-based error term to the conventional ADL framework.

3.1. The synthesis linear classifier

As the dual viewpoint of a commonly used linear classifier $\mathbf{W}$ in $\mathbf{H}\cong\mathbf{W}\mathbf{X}$, the synthesis linear classifier $\mathbf{R}$ takes the following form: $\mathbf{X}\cong\mathbf{R}\mathbf{H}$. It establishes a corresponding relationship between coding coefficients and class labels of data. Thus, the performance of synthesis linear classifier depends on the discriminative ability of input sparse code features. Hence, the introduced classification error term can be described as

$$\min_{\mathbf{R}} \ \|\mathbf{X}-\mathbf{R}\mathbf{H}\|_F^2, \qquad (5)$$

where $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]\in\Re^{m\times N}$ is the sparse coding coefficient matrix. $\mathbf{H}=[\mathbf{h}_1,\mathbf{h}_2,\ldots,\mathbf{h}_N]\in\Re^{K\times N}$ is the label matrix of training data. K is the number of classes in the database. The ith column vector of $\mathbf{H}$ is $\mathbf{h}_i=[0,0,\ldots,1,\ldots,0,0]^T\in\Re^K$, where the location of the only one non-zero element stands for the class of the ith sample. For example, given data matrix $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_6]$ and corresponding label matrix $\mathbf{H}=[\mathbf{h}_1,\mathbf{h}_2,\ldots,\mathbf{h}_6]$, if $\mathbf{y}_1$ is from class 1, $\mathbf{y}_2$ and $\mathbf{y}_3$ are from class 2, and $\mathbf{y}_4$, $\mathbf{y}_5$ and $\mathbf{y}_6$ are from class 3, $\mathbf{H}$ should be

$$\begin{bmatrix}1&0&0&0&0&0\\0&1&1&0&0&0\\0&0&0&1&1&1\end{bmatrix}.$$

For ADL, the length of each sparse code is often larger than the dimensionality of the data. From the theoretical perspective of model (5), it is not necessary to extend $\mathbf{H}$. However, we consider that, if $\mathbf{H}$ is not extended, the label of a testing sample is determined by only one element's location in $\mathbf{h}$ in testing step, which leads to wrong discrimination more easily. So we extend $\mathbf{H}$ by a Kronecker product to improve its robustness. To further strengthen the robustness, we will consider employ more complex coding schemes to extend $\mathbf{H}$ in future, such as the error correction code. For convenience, we make $\mathbf{H}$ have the same number of rows with $\mathbf{X}$ in our paper. In this situation, $\mathbf{R}$ is a square matrix.

The Kronecker product of two arbitrary matrixes $\mathbf{A}$ and $\mathbf{B}$ is usually denoted by $\mathbf{A}\otimes\mathbf{B}$. Given $\mathbf{A}\in\Re^{a\times b}$ and $\mathbf{B}\in\Re^{c\times d}$, $\mathbf{A}\otimes\mathbf{B}$ is an $ac\times bd$ block matrix:

$$\begin{bmatrix}\mathbf{A}_{11}\mathbf{B}&\cdots&\mathbf{A}_{1b}\mathbf{B}\\\vdots&\ddots&\vdots\\\mathbf{A}_{a1}\mathbf{B}&\cdots&\mathbf{A}_{ab}\mathbf{B}\end{bmatrix}.$$

For simplicity, we use Kronecker product of original $\mathbf{H}_0$ and an all ones vector to replace $\mathbf{H}_0$, where the length of the all ones vector is extended multiples of $\mathbf{H}_0$'s dimension. For instance, given an all ones vector $\mathbf{b}=[1,1]^T$, the dimensionality of $\mathbf{H}$ in the above example can be extended. (The aforementioned original $\mathbf{H}$ is rewritten as $\mathbf{H}_0$ in this situation.)

$$\mathbf{H}=\mathbf{H}_0\otimes\mathbf{b}=\begin{bmatrix}1&0&0&0&0&0\\1&0&0&0&0&0\\0&1&1&0&0&0\\0&1&1&0&0&0\\0&0&0&1&1&1\\0&0&0&1&1&1\end{bmatrix}$$

The new $\mathbf{H}$ still includes discriminative information.

3.2. The proposed SLC-ADL model

To make the ADL framework applicable for pattern classification, we incorporate a discriminative error term (5) to the objective function in (4). The optimization problem of our pro-

posed SLC-ADL model can be described as

$$\begin{aligned} \min_{\mathbf{X},\mathbf{\Omega},\mathbf{R}} \quad & \|\mathbf{X}-\mathbf{\Omega Y}\|_F^2+\alpha\|\mathbf{X}-\mathbf{RH}\|_F^2 \\ s.t. \quad & \mathbf{\Omega}\in\mathbf{\Gamma}, \\ & \|\mathbf{x}_i\|_0\le T, \quad i=1,2,\ldots,N, \end{aligned} \tag{6}$$

where $\|\mathbf{X}-\mathbf{RH}\|_F^2$ is the error term to promote the discriminability of ADL. Parameter α is a scalar constant controlling the contribution of $\|\mathbf{X}-\mathbf{RH}\|_F^2$. The set $\mathbf{\Gamma}$ can be constrained to unity row-wise norm to avoid a trivial solution and minimizing the Frobenius norm of $\mathbf{\Omega}$ for stable solution [25]. The sparse code matrix $\mathbf{X}$, analysis dictionary $\mathbf{\Omega}$, and synthesis linear classifier $\mathbf{R}$ can be obtained by solving the optimization problem in (6). Notably, the model (6) is dual to the D-KSVD algorithm [11], where ADL is dual to SDL and SLC is dual to the conventional linear classifier.

3.3. Optimization of SLC-ADL

In recent years, some new dictionary learning optimization strategies have been proposed, such as alternating direction method [34] and half-quadratic-based iterative minimization [35]. In this paper, to learn the analysis dictionary and the synthesis linear classifier efficiently, we develop an alternating iterative algorithm to solve our SLC-ADL model. At first, we initialize the sparse coding coefficients $\mathbf{X}$, and then alternatively update $\{\mathbf{\Omega}, \mathbf{R}\}$ and $\mathbf{X}$. All the terms in model (6) are characterized by Frobenius norm, so this model can be easily solved.

The optimization can be alternately implemented between the following two steps:

(1) Fix $\mathbf{X}$, update $\mathbf{\Omega}$ and $\mathbf{R}$.
According to [25], we can let the constraint set $\mathbf{\Gamma}$ in (4) be a set of matrixes with relatively small Frobenius norm and row-wise norm to be unity. Thus, we firstly have

$$\mathbf{\Omega}^*=arg\min_{\mathbf{\Omega}}\|\mathbf{X}-\mathbf{\Omega Y}\|_F^2+\beta\|\mathbf{\Omega}\|_F^2, \tag{7}$$

where β is the scalar parameter. The penalty term $\|\mathbf{\Omega}\|_F^2$ is one constraint for stable solution. Then, we renormalize each row of $\mathbf{\Omega}$ to unit norm to avoid a trivial solution. Compared with (6), $\|\mathbf{X}-\mathbf{RH}\|_F^2$ is omitted since it has no impact on the sub-optimization of $\mathbf{\Omega}$. Similarly,

$$\mathbf{R}^*=arg\min_{\mathbf{R}}\|\mathbf{X}-\mathbf{RH}\|_F^2. \tag{8}$$

Setting the first order derivative of objective function in (7) to zero, the closed-form solution of $\mathbf{\Omega}$ can be easily obtained as

$$\mathbf{\Omega}^*=\mathbf{XY}^T(\mathbf{YY}^T+\beta\mathbf{I})^{-1}. \tag{9}$$

After renormalizing each row of $\mathbf{\Omega}$ to unit norm, one update of $\mathbf{\Omega}$ is accomplished. In the same way, the closed-form solution of $\mathbf{R}$ is:

$$\mathbf{R}^*=\mathbf{XH}^T(\mathbf{HH}^T+\gamma\mathbf{I})^{-1}, \tag{10}$$

where γ, set to 1e−6, is usually a small number to ensure that the inverse of the matrix $\mathbf{HH}^T$ is solvable. $\mathbf{I}$ is the identity matrix with proper size.

(2) Fix $\mathbf{\Omega}$ and $\mathbf{R}$, update $\mathbf{X}$.
The solution of coding coefficients $\mathbf{X}$ can be obtained in a visual way by equivalent transformation of (6) with respect to $\mathbf{X}$. The transformation is as follow:

$$\begin{aligned} \mathbf{X}^* &= arg\min_{\mathbf{X}} \|\mathbf{X}-\mathbf{\Omega Y}\|_F^2+\alpha\|\mathbf{X}-\mathbf{RH}\|_F^2 \\ &= arg\min_{\mathbf{X}} tr[\mathbf{X}-\mathbf{\Omega Y})(\mathbf{X}-\mathbf{\Omega Y})^T] \\ &\quad +\alpha tr[(\mathbf{X}-\mathbf{RH})(\mathbf{X}-\mathbf{RH})^T] \\ &= arg\min_{\mathbf{X}} (1+\alpha)tr(\mathbf{XX}^T)-2tr[(\mathbf{\Omega Y}+\alpha\mathbf{RH})\mathbf{X}^T] \\ &= arg\min_{\mathbf{X}} tr[\mathbf{XX}^T-\frac{2}{1+\alpha}(\mathbf{\Omega Y}+\alpha\mathbf{RH})\mathbf{X}^T] \\ &= arg\min_{\mathbf{X}} \left\|\mathbf{X}-\tfrac{1}{\alpha+1}(\mathbf{\Omega Y}+\alpha\mathbf{RH})\right\|_F^2 \\ &\quad s.t. \quad \|x_i\|_0\le T, \quad i=1,2,\ldots,N, \end{aligned} \tag{11}$$

where the optimal solution will not be affected by the constant term without relation to $\mathbf{X}$. For convenient representation, we define a thresholding function operator

$$\mathbf{A}^*=HT(\mathbf{A},z), \tag{12}$$

where $HT(\mathbf{A}, z)$ is a non-linear function retaining the z largest entries (in magnitude) in each column vector of $\mathbf{A}$ and setting the others to 0. According to the final form in (11), it is easy to be seen that the optimalizing solution of $\mathbf{X}$ can be described as

$$\mathbf{X}^*=HT\left(\frac{\mathbf{\Omega Y}+\alpha\mathbf{RH}}{1+\alpha},T\right). \tag{13}$$

The above thresholding function operation makes the T largest numbers (in magnitude) unchanged in each column vector of $\frac{\mathbf{\Omega Y}+\alpha\mathbf{RH}}{1+\alpha}$ and sets the others to 0. The obtained result is the optimal coding coefficient matrix $\mathbf{X}^*$.

The above update steps are summarized in Algorithm 1.

Algorithm 1. SLC-ADL algorithm.

Input:
Training data $\mathbf{Y}$ and corresponding label matrix $\mathbf{H}$;
Parameters α, β, γ, and T.

Output:
The analysis dictionary $\mathbf{\Omega}$;
The synthesis linear classifier $\mathbf{R}$.

1: Set $\mathbf{\Omega}^{(0)}$ and $\mathbf{R}^{(0)}$ as random matrix with proper size, $\mathbf{X}^{(0)}=\mathbf{H}$ for initialization, $t=0$;
2: **while** not convergence **do**
3: $t\leftarrow t+1$;
4: Update $\mathbf{\Omega}^{(t)}$ according to (9) followed by renormalizing $\mathbf{\Omega}^{(t)}$ row-wise;
5: Update $\mathbf{R}^{(t)}$ according to (10);
6: Update $\mathbf{X}^{(t)}$ according to (13);
7: **end while**

3.4. Convergence analysis

Since optimal solution formulas of updating $\mathbf{X}$, $\mathbf{\Omega}$ and $\mathbf{R}$ have been solved in Section 3.3, the alternative iteration algorithm between $\{\mathbf{\Omega}, \mathbf{R}\}$ and $\mathbf{X}$ can be performed. Assuming that, in the tth iteration, $\mathbf{X}$, $\mathbf{\Omega}$ and $\mathbf{R}$ are indicated as $\mathbf{X}^t$, $\mathbf{\Omega}^t$, and $\mathbf{R}^t$ separately. In the (t+1)th iteration step, $\mathbf{\Omega}^{t+1}$ and $\mathbf{R}^{t+1}$ can be obtained by $\mathbf{X}^t$, and $\mathbf{X}^{t+1}$ can be updated by $\mathbf{\Omega}^{t+1}$ and $\mathbf{R}^{t+1}$.

Let $F(\mathbf{X}, \mathbf{\Omega}, \mathbf{R})$ stand for the minimum value of the objective function in (6).

$$\begin{aligned} F(\mathbf{X}^t,\mathbf{\Omega}^t,\mathbf{R}^t) &= \min_{\mathbf{X}^t,\mathbf{\Omega}^t,\mathbf{R}^t}\|\mathbf{X}^t-\mathbf{\Omega}^t\mathbf{Y}\|_F^2+\alpha\|\mathbf{X}^t-\mathbf{R}^t\mathbf{H}\|_F^2, \\ F(\mathbf{X}^{t+1},\mathbf{\Omega}^{t+1},\mathbf{R}^{t+1}) &= \min_{\mathbf{X}^{t+1},\mathbf{\Omega}^{t+1},\mathbf{R}^{t+1}}\|\mathbf{X}^{t+1}-\mathbf{\Omega}^{t+1}\mathbf{Y}\|_F^2+\alpha\|\mathbf{X}^{t+1} \\ &\quad -\mathbf{R}^{t+1}\mathbf{H}\|_F^2. \end{aligned} \tag{14}$$

When fixing $\mathbf{X}^t$, $\mathbf{\Omega}^{t+1}$ is updated according to its analytical solution, which is solved by minimization problem (7). Thus, the object function will decrease in this step. In this step, we can obtain

$$F(\mathbf{X}^t, \mathbf{\Omega}^{t+1}, \mathbf{R}^{t+1}) \leq F(\mathbf{X}^t, \mathbf{\Omega}^t, \mathbf{R}^t). \tag{15}$$

Then, $\mathbf{R}^{t+1}$ is also updated according to an analytical solution obtained by solving minimization problem (8). We can get the following inequality

$$F(\mathbf{X}^t, \mathbf{\Omega}^{t+1}, \mathbf{R}^{t+1}) \leq F(\mathbf{X}^t, \mathbf{\Omega}^{t+1}, \mathbf{R}^t). \tag{16}$$

When updating the coding coefficient $\mathbf{X}$, it can be solved by optimizing (11). Eq. (11) aims to obtain the solution of minimizing the terms related to $\mathbf{X}$ in (6). In this solving process, the thresholding function can be effectively solved according to the Definition 4.1 in [29]. Because of the properties of semi-algebraic functions, the thresholding operator will not damage the overall downside trend. Therefore, the objective will monotonously decrease to a convergent solution. Hence, the following inequality is true,

$$F(\mathbf{X}^{t+1}, \mathbf{\Omega}^{t+1}, \mathbf{R}^{t+1}) \leq F(\mathbf{X}^t, \mathbf{\Omega}^{t+1}, \mathbf{R}^{t+1}). \tag{17}$$

Combining the inequalities (15)–(17), we have

$$\begin{aligned} F(\mathbf{X}^{t+1}, \mathbf{\Omega}^{t+1}, \mathbf{R}^{t+1}) &\leq F(\mathbf{X}^t, \mathbf{\Omega}^{t+1}, \mathbf{R}^{t+1}) \leq F(\mathbf{X}^t, \mathbf{\Omega}^{t+1}, \mathbf{R}^t) \\ &\leq F(\mathbf{X}^t, \mathbf{\Omega}^t, \mathbf{R}^t), \end{aligned} \tag{18}$$

which intuitively indicates that the objective function of (6) is non-increasing in terms of energy until convergence.

The convergence of $F(\mathbf{X}^t, \mathbf{\Omega}^t, \mathbf{R}^t)$ cannot ensure the convergence of variables $\{\mathbf{X}^t, \mathbf{\Omega}^t, \mathbf{R}^t\}$. To further analyze the convergence of the variables, we need recur to Theorem 4.9 in [36]. According to Theorem 4.9, we can obtain two conclusions.

Conclusion 1: The sequence of $\{\mathbf{X}^t, \mathbf{\Omega}^t, \mathbf{R}^t\}$ generated by our optimization algorithm has at least one accumulation point and all the accumulation points are partial optima of objective function and have the same function value.

Conclusion 2: If the subproblem of $\{\mathbf{\Omega}, \mathbf{R}\}$ has a unique solution, then the sequence $\{\mathbf{X}^t, \mathbf{\Omega}^t, \mathbf{R}^t\}$ generated by our optimization algorithm satisfies:

$$\lim_{t\to\infty} \|\mathbf{X}^{t+1} - \mathbf{X}^t\| + \|\mathbf{\Omega}^{t+1} - \mathbf{\Omega}^t\| + \|\mathbf{R}^{t+1} - \mathbf{R}^t\| = 0. \tag{19}$$

When omitting the sparsity constraint of l_0-norm, the two conclusions can be easily proved by referring to the properties of bi-convex problem [36] and alternative convex search (ACS) algorithm [37]. After adding the l_0-norm constraint, the optimization problem become non-convex. In the process of optimization, the thresholding function of l_0-norm can be realized according to the Definition 4.1 in [29]. This operation retains some largest entries (in magnitude) in each column vector of $\mathbf{X}$ and sets the others to 0. Because of the properties of semi-algebraic functions, it will not damage the overall downside trend of objective function. In addition, generally speaking, this operation will not influence the uniqueness of solution in each iteration. In this case, condition 2 in Theorem 4.9 [36] that the X subproblem in (11) has a unique solution is still satisfied, which leads to Conclusion 1. Having obtained Conclusion 1, if $\{\mathbf{\Omega}, \mathbf{R}\}$ also has the unique optimal solution in their subproblem, Conclusion 2 can be obtained based on condition 3 in Theorem 4.9 [36]. Therefore, the above two conclusions can still be obtained when adding the l_0-norm constraint. Combining the monotone convergence of objective function and the above two conclusions about the variables, we show that our optimization algorithm usually converges to the local optimum solution and may converge to the global optimum solution.

Fig. 1 demonstrates the convergence curves of our proposed algorithm on several widely used databases including Scene 15, Caltech 101, UCF 50, Extended YaleB and AR. They verify the convergence in practice.

3.5. Computational complexity

In training stage, the efficiency improvement mainly benefits from the rapid convergence rate. Then, the variations $\mathbf{\Omega}$ and $\mathbf{R}$ have the closed-solution easily solved, which is significantly superior to iterative solving strategies. Computing $\mathbf{\Omega}^* = \mathbf{X}\mathbf{Y}^T(\mathbf{Y}\mathbf{Y}^T + \beta\mathbf{I})^{-1}$ needs $O(nmN + n^3)$ operations with pre-computed $\mathbf{Y}\mathbf{Y}^T$. The unit row-wise norm will cost time $O(mn)$, which can be neglected. Similarly, the time complexity of $\mathbf{R}^* = \mathbf{X}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T + \gamma\mathbf{I})^{-1}$ is $O(nmN + m^3)$ with pre-computed $\mathbf{H}\mathbf{H}^T$. The process of solving $\mathbf{X}$ has a complexity of $O(nmN)$, where the thresholding operation can be done in $O(mN)$ time [29]. As a whole, the overall complexity of training is about $O(mnN + m^3 + n^3)$ in one iteration.

4. Classification schemes

Given training data $\mathbf{Y}$ and its label matrix $\mathbf{H}$, we can apply SLC-ADL algorithm to learn the analysis dictionary $\mathbf{\Omega}$ and the synthesis linear classifier $\mathbf{R}$. Both the learned $\mathbf{\Omega}$ and $\mathbf{R}$ are crucial in classification schemes. $\mathbf{\Omega}$ is used to project the original data $\mathbf{Y}$ to coding coefficient features. $\mathbf{R}$ is utilized to obtain the class labels of data.

4.1. Minimizing the classification error

By $\mathbf{x} = HT(\mathbf{\Omega}\mathbf{y}, s)$, we obtain the coding coefficient $\mathbf{x}_{test}$ of $\mathbf{y}_{test}$. Then we design a standard label matrix $\mathbf{L}_0 \in \Re^{K\times K}$. It is a unit matrix and its size K is the total number of classes in the database. While the length of each sparse code is larger than the dimensionality of data, $\mathbf{L}_0$ would be turned to $\mathbf{L}$ by Kronecker product to increase its dimensionality. Each column of the desired $\mathbf{L}$ still contains discrimination. Based on the error term induced by the synthesis linear classifier, we can contrast the code $\mathbf{x}_{test}$ with the product of $\mathbf{R}$ and each column $\mathbf{l}_j$, $j = 1, 2, \ldots, K$ in $\mathbf{L}$, respectively. Our goal is to find the column vector $\mathbf{l}_{predict}$ to make the error minimum. This classification scheme can be described as

$$\mathbf{l}_{predict} = arg\min_{\mathbf{l}_j} \|\mathbf{x}_{test} - \mathbf{R}\mathbf{l}_j\|_2^2, \quad j = 1, 2, \ldots, K. \tag{20}$$

The class label of $\mathbf{y}_{test}$ can be derived from $\mathbf{l}_{predict}$.

4.2. *H*-space based kNN classifier

k Nearest Neighbor (kNN) [38] is a classical and simple classification method. It does not require training effort and mainly depends on the distance metrics among samples. The classification of a test sample is determined by the category labels of its k nearest training samples. As a simple method, kNN is often studied by researchers to improve its performance [39,40]. It is also widely used in some practical applications [41] or serves as a basic component in many complex models [42,43].

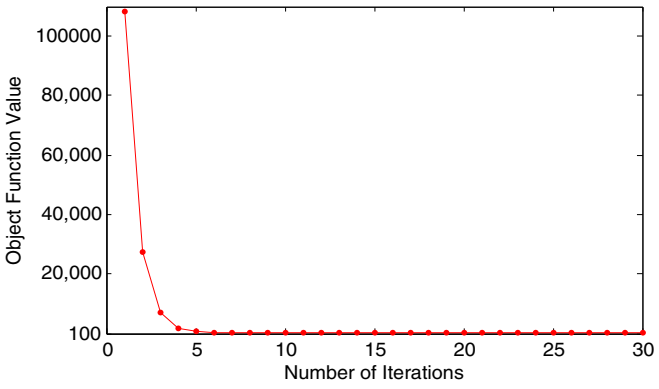
In our classification step, we attempt to use kNN to classify the testing samples in new $\mathbf{H}$-space and set k to 1. To acquire the coding coefficients, we apply $\mathbf{x} = HT(\mathbf{\Omega}\mathbf{y}, s)$ on training and testing data at the same time. After that, we minimize the classification error term (17) to obtain the label vectors of training and testing data

$$\mathbf{h} = arg\min_{\mathbf{h}} \|\mathbf{x} - \mathbf{R}\mathbf{h}\|_2^2, \tag{21}$$

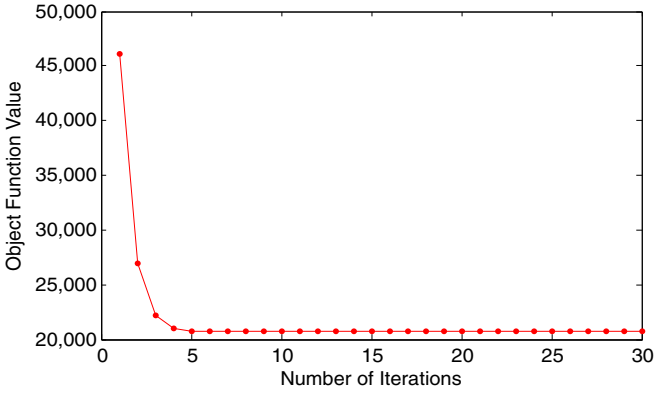
where $\mathbf{R}$ and $\mathbf{x}$ are known. It can be seen that this is a standard least squares problem and it has a closed-form solution, so we can easily obtain the label matrix by

$$\mathbf{h} = (\mathbf{R}^T\mathbf{R} + \gamma\mathbf{I})^{-1}\mathbf{R}^T\mathbf{x}. \tag{22}$$

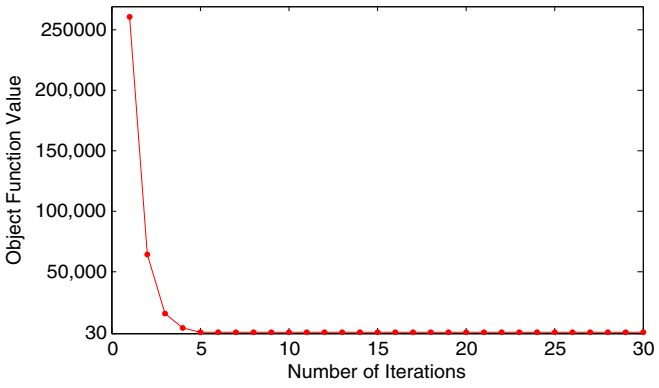
Once the class labels of training data and testing data are obtained, we regard the label vectors as new features and utilize kNN to perform classification in the $\mathbf{H}$-space.



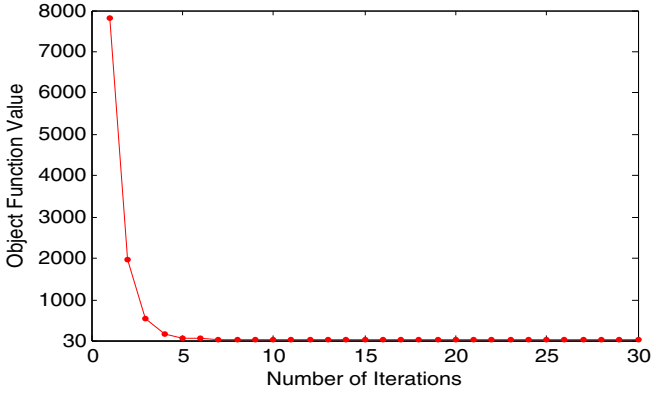
(a) Scene 15 database



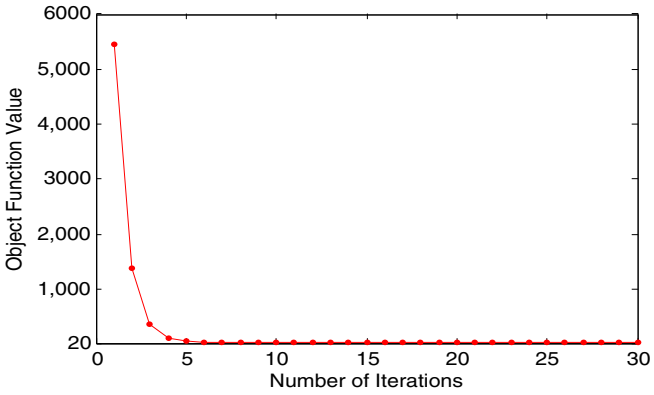
(b) Caltech 101 database



(c) UCF 50 database



(d) Extended YaleB database



(e) AR database

Fig. 1. The convergence curves of SLC-ADL on five common databases.

4.3. Employing the nearest base in L

For **H**-space based *k*NN classifier, if the database is large scale, the process of classification has a heavy calculation burden and is time-consuming. To reduce the computational burden, we design another new classification scheme.

In this scheme, we predefine a desired label matrix **L** (as introduced in Section 4.1) instead of solving the training data's label matrix. So we only need to solve each testing sample's label vector in the classification stage, and then contrast each obtained label vector with the column vectors of **L** to determine the final category of each testing sample.

Mapping the testing sample $\mathbf{y}_{test}$ to $\mathbf{x}_{test}$ by $\mathbf{x} = HT(\mathbf{\Omega y}, s)$, where s is an adjustable sparsity parameter in classification step. Then, we employ formulation (18) to obtain the analytical solution of each testing sample's label vector. The category of a testing sample is determined by the label of its label vector's nearest base in $\mathbf{L}$.

4.4. Discussion

Classification scheme 1 is based on the error in the transform domain. Classification scheme 2 uses a kNN classifier directly in the $\mathbf{H}$-space after obtaining the vector $\mathbf{h}$ of corresponding testing sample. To reduce the computational burden of classification scheme 2, we design the classification scheme 3. It predefines a desired label matrix instead of solving the training data's label matrix, which raises efficiency. When the database is large scale, the computation of Eq. (18) is time-consuming. Hence, the classification scheme 1 and classification scheme 3 are more suitable for large scale database. Compared to classification scheme 3, the testing efficiency of scheme 1 is higher because it does not need to compute the label of testing data according to Eq. (18). The experiment results in Section 5 will verify this property.

5. Experiments

In this section, we perform experiments on five benchmark datasets to evaluate the performance of SLC-ADL. They are the scene categorization dataset: Scene 15 [44], object classification dataset: Caltech 101 [45], action dataset: UCF 50 [46], and two face databases: Extended YaleB [47] and AR face [48]. The above datasets are widely used to verify DL algorithms in previous works. We use the features provided by Jiang [17][1] and Corso [49].[2]

At first, to verify the advantages of SLC-ADL, we choose state-of-the-art DL algorithms to make comparisons. They are synthesis-dictionary-based algorithms, such as the Collaborative-Representation-based Classifier (CRC) [50], the classical SRC [1], DLSI [19], FDDL [20], LC-KSVD [17], and the recently proposed projective Dictionary Pair Learning (DPL) [27]. Besides the classification accuracy, we also record the training and testing time of these competing algorithms in experiments.

Then, we offer comparison experiments to verify the advantages of SLC. we compare our SLC-ADL with the ADL + Support Vector Machine (SVM) [25], ADL + Nearest Neighbor (NN) and ADL + conventional Analysis Linear Classifier (ALC) as described in expression (18).

All the experiments are implemented in Matlab of the same computer with 32 GB memory and 2.6 GHz Intel CPU. To keep the comparisons fair, we adopt the experiment settings in [17] and [27] for all the competing methods. In the classification stage, our proposed SLC-ADL uses three different classification schemes. For convenience, we refer to them as SLC-ADL1, SLC-ADL2, and SLC-ADL3, respectively.

5.1. Comparison with SDL methods

5.1.1. Scene categorization

In Scene 15 database [44], there are 200–400 images with approximate average size of 250 × 300 pixels in each category. The fifteen scenes contain MIThighway, bedroom, kitchen, and so on, as shown in Fig. 2. Following the experimental settings in [17], we choose 100 images per category as training data and the remaining as testing data. In this paper, we adopt the spatial pyramid features reduced to 3000-d provided by [17] to perform the experiment. To make the length of each sparse code larger than the dimensionality of data, we extend the label matrix to 3015-d by Kronecker product between the original label matrix and a 201-d all ones vector. The parameter α is set to 50 in three classification schemes and β is set to 0.1 in SLC-ADL1 and 0.5 in SLC-ADL2 and SLC-ADL3, which are obtained by cross validation.

Fig. 2. The typical examples from Scene 15 database.

Fig. 3. The typical examples from Caltech 101 database.

Table 1
Results on the Scene 15 database.

	Accuracy (%)	Training time (s)	Testing time (s)
CRC	92.0	No need	1.82e −2
SRC	91.8	No need	0.63
DLSI	91.7	3616.4	0.69
FDDL	92.3	4701.5	0.80
LC-KSVD	92.9	137.8	3.10e−4
DPL	97.7	27.72	4.10e−4
SLC-ADL1	**98.2**	38.10	1.98e−4
SLC-ADL2	97.1	113.22	5.73e−3
SLC-ADL3	98.1	113.56	6.93e−4

The comparative results of all methods are demonstrated in Table 1. From the experimental results, it can be seen that our proposed SLC-ADL1 has the best performance in both accuracy and testing efficiency, which verifies the validity of our algorithm in scene categorization tasks.

5.1.2. Object classification

We use Caltech 101 database [45] to evaluate our SLC-ADL algorithm on object classification problems. There are 9144 images from one background class and 101 object classes in Caltech 101 database, which includes anchors, soccer balls, cups, and so on, like Fig. 3. In each class, the number of samples varies from 31 to 800. According to the experimental settings in [17] and [44], we randomly select 30 samples per category for training and the remainder for testing. The features used in our experiment are extracted by the standard Bag-of-Words (BOW) + Spatial Pyramid Matching (SPM). We use dense SIFT descriptors to calculate the SPM feature, which are extracted on 1 × 1, 2 × 2, and 4 × 4 size grids. Then, a vector quantization based coding method is applied to extract mid-level features and the standard max pooling approach is used to obtain high-dimensioned pooled features. Finally, we employ

[1] http://www.umiacs.umd.edu/~zhuolin/projectlcksvd.html.
[2] http://www.cse.buffalo.edu/~jcorso/r/actionbank.

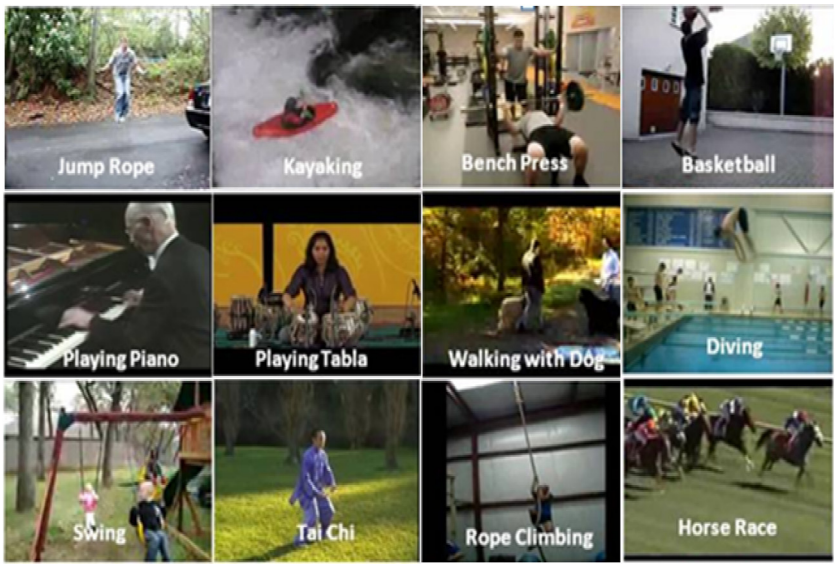


Fig. 4. The typical examples from UCF 50 database.

Table 2
Results on the Caltech 101 database.

	Accuracy (%)	Training time (s)	Testing time (s)
CRC	68.2	No need	2.06e−2
SRC	70.7	No need	15.18
DLSI	73.1	11561	5.64
FDDL	73.2	104050	13.10
LC-KSVD	73.6	3121.56	1.99e−3
DPL	73.9	149.80	2.53e−3
SLC-ADL1	**76.4**	177.24	3.06e−4
SLC-ADL2	75.6	28.13	1.07e−2
SLC-ADL3	75.7	169.59	7.16e−4

Table 3
Results on the UCF 50 database.

	Accuracy (%)	Training time (s)	Testing time (s)
CRC	75.6	No need	6.43e−2
SRC	75.0	No need	2.25
DLSI	75.4	168500	6.40
FDDL	76.6	219050	51.16
LC-KSVD	70.1	11811	3.64e−3
DPL	77.4	308.33	2.68e−3
SLC-ADL1	77.8	126.27	8.65e−4
SLC-ADL2	**78.3**	731.54	3.67e−2
SLC-ADL3	77.4	122.15	3.22e−3

Principal Component Analysis (PCA) to reduce the original 21,504-d samples to 3000-d. To ensure the length of each sparse code being larger than the dimensionality of data, the label vectors are extended to 3060-d by Kronecker product. The parameters α and β are 50 and 5 in all three approaches.

The comparative results of all methods are listed in Table 2. For Caltech 101 dataset, SLC-ADL1 has the highest classification accuracy 76.4%, and has a 2.5% improvement over DPL and a 2.8% improvement over other synthesis-dictionary-based competing methods. It should also be noticed that SLC-ADL1 has a very high efficiency especially in testing stage since the coding coefficients can be obtained by a simple matrix multiplication and a thresholding function operation. The results verify our discussion in Section 4.4.

5.1.3. Action recognition

The classification experiment on UCF 50 database [46] is a challenging action recognition task, which contains 6680 human action videos of 50 categories. All these videos are chosen from YouTube, as shown in Fig. 4. We use the action bank features provided by [49] to evaluate our SLC-ADL algorithm. In each category, samples are divided into five folds, and four of them are randomly chosen as training data and the rest as testing data. To compare all the algorithms fairly, we reduce the dimension of features to 5000 by PCA for all the methods. To make the length of each sparse code larger than the dimensionality of data, the label matrix is extended to 5050-d. By cross validation, α is set to 50 in three SLC-ADLs and β is tuned as 5e−3 in SLC-ADL1 and SLC-ADL3, and 5e−2 in SLC-ADL2.

The results of different competing methods are demonstrated in Table 3. Although SLC-ADL2 achieves the highest classification accuracy, its running efficiency is not satisfactory due to the large scale of UCF 50 database. In contrast, SLC-ADL1 and SLC-ADL3 are superior to DPL in terms of testing speed. Meanwhile, their accuracies are not less than these state-of-the-art algorithms.

5.1.4. Face recognition

We choose Extended YaleB [47] and AR [48], two widely used face datasets to evaluate our SLC-ADL algorithm. The Extended YaleB database includes 2414 front face images of 38 persons, which have big differences in expressions and illumination, as shown in Fig. 5(a). These images are cropped to 192 × 168 pixels. Then, by a randomly generated matrix, the image pixels are projected to 504-d vectors as random-face features. For Extended YaleB dataset, random half of the images per category are for training and the rest half are for testing. To make the length of each sparse code larger than the dimensionality of data, the label matrix is extended to 570-d.

The AR database contains expressions, illumination, and occlusions variations, as illustrated in Fig. 5(b). Similar to Extended YaleB, the original images from AR database are cropped to 165 × 120, and then projected to 540-d. We follow the experimental settings of Jiang's work [17]. There are 2600 samples from 50 males and 50 females, and for each category, random 20 samples are selected for training and the other for testing. To ensure the length of each sparse code being larger than the dimensionality of data, the label matrix is extended to 600-d. The features of Extended YaleB database and AR database are provided

(a) Extended YaleB database (b) AR database

Fig. 5. The typical examples from Extended YaleB database and AR database.

Table 4
Results on the Extended YaleB database.

	Accuracy (%)	Training time (s)	Testing time (s)
CRC	97.0	No need	2.25e−3
SRC	96.5	No need	3.40e−2
DLSI	97.0	38.03	1.48e−2
FDDL	96.7	418.39	0.49
LC-KSVD	96.7	63.69	3.35e−4
DPL	97.5	4.97	1.82e−4
SLC-ADL1	96.7	4.29	4.41e−5
SLC-ADL2	**97.8**	0.70	9.74e−4
SLC-ADL3	96.2	0.69	6.10e−5

Table 5
Results on the AR database.

	Accuracy (%)	Training time (s)	Testing time (s)
CRC	98.0	No need	5.77e−3
SRC	97.5	No need	4.97e−2
DLSI	97.5	40.55	2.27e−2
FDDL	97.5	446.7	0.34
LC-KSVD	97.8	85.53	2.99e−4
DPL	98.3	14.69	5.48e−4
SLC-ADL1	97.2	1.09	5.52e−5
SLC-ADL2	**98.6**	1.03	1.04e−3
SLC-ADL3	97.2	1.02	8.86e−5

by Jiang et al. [17]. α on Extended YaleB database and AR database is the same, equal to 50. β on Extended YaleB is set to 0.05 in SLC-ADL1 and SLC-ADL3, 0.5 in SLC-ADL2. However, β on AR is set to 0.01 in SLC-ADL1 and SLC-ADL3, 0.1 in SLC-ADL2.

The experimental results of accuracy and efficiency are listed in Tables 4 and 5. For the two face databases, all the accuracies achieved by the comparative methods are over 96%, so there is no remarkable promotion. Even so, our proposed SLC-ADL2 still realizes the highest accuracy in two datasets. They are 97.8% and 98.6% respectively. What is more, the advantage of running efficiency is obvious, especially for SLC-ADL1 and SLC-ADL3.

5.2. Comparison with conventional ADL methods

In this subsection, we choose the challenging Caltech 101 database [45] to test the advantages of SLC-ADL compared to other ADL models using different classifiers in sparse code domain, which include ADL + Support Vector Machine (SVM) [25], ADL + Nearest Neighbor (NN) and ADL + conventional Analysis Linear Classifier (ALC) as described in expression (18).

For fair comparison, all the classifiers are used on the sparse codes and the results are obtained in their optimal parameters. The comparison results are shown as Table 6. From Table 6, we can see that all the three classification schemes based on the SLC have a competitive accuracy, which proves that our proposed SLC is meaningful and valuable.

Table 6
Comparison with conventional ADL methods on Caltech 101 database.

	ADL+SVM	ADL+NN	ADL+ALC	SLC-ADL1	SLC-ADL2	SLC-ADL3
Accuracy (%)	6 4.5	75.6	72.8	76.4	75.6	75.7

5.3. Result analysis

By analyzing comparison results, the proposed SLC-ADL can achieve higher accuracy than state-of-the-art SDL methods, which indicates that ADL incorporating discrimination can address classification problems well. Compared to the basic ADL model, the better performance of SLC-ADL further proves the validity of the synthesis-linear-classifier based error term. In addition to the accuracy, our proposed method has a satisfactory running efficiency. The good training efficiency mainly benefits from the rapid convergence rate. The testing speeds of the first and third classification scheme are more competitive than SDL methods. The computational complexity of these two classification schemes is dominated by the calculation of obtaining coding coefficients, at $O(mnN)$. In the second classification scheme, computing the label matrix requires another $O(m^3 + m^2N)$ operations. In short, the testing efficiency of ADL method is usually superior to SDL methods due to the easily obtained coding coefficients by matrix multiplication, which reduces the computational complexity.

6. Conclusion

This paper presents a novel discriminative ADL model. We incorporate a synthesis-linear-classifier-based classification error term into the basic ADL model, which fully exploits category information in the learning process to improve discriminability. The algorithm is solved by an iterative method and the obtained solutions in each alternating stage are closed-form, which reduces the computation complexity. Since the coding coefficients can be obtained by a simple matrix multiplication and a thresholding function operation, the testing efficiency is improved largely. Based on the proposed synthesis linear classifier, we design three classification schemes to accomplish pattern classification tasks. The experimental results for scene categorization, object classification, action recognition and face recognition show the advantages of our proposed SLC-ADL in terms of classification accuracy and running efficiency.

References

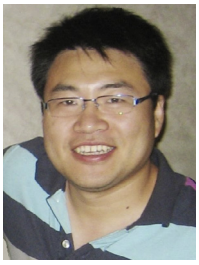
[1] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[2] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, USA, 2008, pp. 1–8.

[3] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 2008, pp. 1794–1801.
[4] J. Yang, M. Yang, Top-down visual saliency via joint CRF and dictionary learning, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 2012, pp. 2296–2303.
[5] Y. Li, Y. Guo, J. Guo, M. Li, X. Kong, CRF with locality-consistent dictionary learning for semantic segmentation, in: Proceedings of Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 2015.
[6] R. He, T.N. Tan, L. Wang, Robust recovery of corrupted low-rank matrix by implicit regularizers, IEEE Trans. Pattern Anal. Mach. Intell. 36 (4) (2014) 770–783.
[7] S.Y. Yang, Z.Z. Liu, M. Wang, F.H. Sun, L.C. Jiao, Multitask dictionary learning and sparse representation based single-image super-resolution reconstruction, Neurocomputing 74 (17) (2011) 3193–3203.
[8] S.P. Zhang, H.X. Yao, H.Y. Zhou, X. Sun, S.H. Liu, Robust visual tracking based on online learning sparse representation, Neurocomputing 100 (2013) 31–40.
[9] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.
[10] K. Huang, S. Aviyente, Sparse representation for signal classification, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), Poznan, Poland, 2006, pp. 609–616.
[11] Q. Zhang, B. Li, Discriminative K-SVD for dictionary learning in face recognition, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 2010, pp. 2691–2698.
[12] M. Yang, L. Zhang, J. Yang, D. Zhang, Robust sparse coding for face recognition, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, IEEE, 2011, pp. 625–632.
[13] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.
[14] Y. Zhou, H. Zhao, L. Shang, T. Liu, Immune K-SVD algorithm for dictionary learning in speech denoising, Neurocomputing 137 (2014) 223–233.
[15] Y. Song, W. Cao, Z.L. He, Robust iris recognition using sparse error correction model and discriminative dictionary learning, Neurocomputing 137 (2014) 198–204.
[16] R. Rubinstein, A.M. Bruckstein, M. Elad, Dictionaries for sparse representation modeling, Proc. IEEE 98 (6) (2010) 1045–1057.
[17] Z.L. Jiang, Z. Lin, L.S. Davis, Label consistent K-SVD: learning a discriminative dictionary for recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2651–2664.
[18] M. Yang, L. Zhang, J. Yang, D. Zhang, Metaface learning for sparse representation based face recognition, in: Proceedings of IEEE International Conference on Image Processing (ICIP), IEEE, 2010, pp. 1601–1604.
[19] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, 2010, pp. 3501–3508.
[20] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher discrimination dictionary learning for sparse representation, in: Proceedings of IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 543–550.
[21] S. Kong, D. Wang, A dictionary learning approach for classification: separating the particularity and the commonality, in: Proceedings of European Conference On Computer Vision (ECCV), Springer, Florence, Italy, 2012, pp. 186–199.
[22] N. Zhou, Y. Shen, J. Peng, J. Fan, Learning inter-related visual dictionary for object recognition, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), IEEE, Providence, RI, USA, 2012, pp. 3490–3497.
[23] R. Rubinstein, T. Peleg, M. Elad, Analysis K-SVD: a dictionary-learning algorithm for the analysis sparse model, IEEE Trans. Signal Process. 61 (3) (2013) 661–677.
[24] S. Ravishankar, Y. Bresler, Learning sparsifying transforms, IEEE Trans. Signal Process. 61 (5) (2013) 1072–1086.
[25] S. Shekhar, V.M. Patel, R. Chellappa, Analysis sparse coding models for image-based classification, in: Proceedings of IEEE International Conference on Image Processing (ICIP), Paris, France, 2014, pp. 5207–5211.
[26] R. Rubinstein, M. Elad, Dictionary learning for analysis-synthesis thresholding, IEEE Trans. Signal Process. 62 (22) (2014) 5962–5972.
[27] S.H. Gu, L. Zhang, W. Zuo, X. Feng, Projective dictionary pair learning for pattern classification, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), Montreal, Canada, 2014, pp. 793–801.
[28] L. Shen, G. Sun, Q.M. Huang, S.H. Wang, Z.C. Lin, E.H. Wu, Multi-level discriminative dictionary learning with application to large scale image classification, IEEE Trans. Image Process. 24 (10) (2015) 3109–3123.
[29] J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Mathematical Programming 146 (1–2) (2014) 459–494.
[30] J. Dong, W.W. Wang, W. Dai, Analysis SimCO: a new algorithm for analysis dictionary learning, in: Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), IEEE, 2014, pp. 7193–7197.
[31] B. Wen, S. Ravishankar, Y. Bresler, Structured overcomplete sparsifying transform learning with convergence guarantees and applications, Int. J. Comput. Vis. 114 (2–3) (2015) 137–167.
[32] S. Ravishankar, Y. Bresler, Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging, SIAM J. Imag. Sci. 8 (4) (2015) 2519–2557.
[33] E. Eksioglu, O. Bayir, K-SVD meets transform learning: transform K-SVD, IEEE Trans. Signal Process. 21 (3) (2014) 347–351.
[34] A. Rakotomamonjy, Applying alternating direction method of multipliers for constrained dictionary learning, Neurocomputing 106 (2013) 126–136.
[35] R. He, W. Zheng, T. Tan, Z. Sun, Half-quadratic-based iterative minimization for robust sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 261–275.
[36] J. Gorski, F. Pfeuffer, K. Klamroth, Biconvex sets and optimization with biconvex functions: a survey and extensions, Math. Methods Oper. Res. 66(3) (2007) 373–407.
[37] R.E. Wendell, J.A.P. Hurter, Minimization of a non-separable objective function subject to disjoint constraints, Oper. Res. 24 (4) (1976) 643–657.
[38] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[39] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 18 (6) (1996) 607–616.
[40] C. Domeniconi, D. Gunopulos, J. Peng, Large margin nearest neighbor classifiers, IEEE Trans. Neural Netw. 16 (4) (2005) 899–909.
[41] H.X. He, S. Hawkins, W. Graco, X. Yao, Application of genetic algorithm and k-nearest neighbour method in real world medical fraud detection problem, J. Adv. Comput. Intell. 4 (2) (2000) 130–137.
[42] N. Nguyen, Y.S. Guo, Metric learning: a support vector approach, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2008, pp. 125–136.
[43] S. McCann, D.G. Lowe, Local naive Bayes nearest neighbor for image classification, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 3650–3656.
[44] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), New York, NY, vol. 2, 2006, pp. 2169–2178.
[45] F.F. Li, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, Comput. Vis. Image Underst. 106 (1) (2007) 59–70.
[46] K.K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, Mach. Vis. Appl. 24 (5) (2013) 971–981.
[47] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.
[48] A. Martinez, R. Benavente, The AR face database, CVC Technical Report 24, 1998.
[49] S. Sadanand, J.J. Corso, Action bank: A high-level representation of activity in video, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), Providence, RI, 2012, pp. 1234–1241.
[50] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: Which helps face recognition? in: Proceedings of IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 471–478.

Jiujun Wang received the B.E. degree in English Intensive, Electronic and Information Engineering, Dalian University of Technology, in 2015. She is currently a Master Student in the School of Information and Communication Engineering, Dalian University of Technology. Her research interests are in dictionary learning and machine learning.

Yanqing Guo received the B.S. degree and Ph.D. degree in Electronic Engineering from Dalian University of Technology of China, in 2002 and 2009, respectively. He is currently an Associate Professor with Faculty of Electronic Information and Electrical Engineering Dalian University of Technology. His research interests include multimedia security and forensics, digital image processing and machine learning.

Jun Guo received the B.S. degree in electronics and information engineering and the M.S. degree in information and communication engineering from Dalian University of Technology of China, in 2013 and 2016, respectively. He is currently a Ph.D. candidate in Tsinghua-Berkeley Shenzhen Institute, Tsinghua University. His research interests include pattern recognition and machine learning. In particular, he focuses on dictionary learning, matrix factorization and their applications on multimedia and data processing.



Ming Li received the M.S. and Ph.D. degrees in electrical engineering from the State University of New York at Buffalo, Buffalo, NY, USA, in 2005 and 2010, respectively. He is currently an Associate Professor with Faculty of Electronic Information and Electrical Engineering Dalian University of Technology. His research interests are in the general areas of communication theory and signal processing with applications to interference channels and signal waveform design, secure wireless communications, cognitive radios and networks, data hiding and steganography, and compressed sensing.

Xiangwei Kong is a Professor of Department of Electronic and Information Engineering, and Vice Director of Information Security Research Center of Dalian University of Technology, Dalian, China. She received B.E. and M.Sc. degree from Harbin Shipbuilding Engineering Institute in 1985 and 1988 and received Ph.D. degree from Dalian University of Technology, in 2003. Her research interests include multimedia security and forensics, digital image processing and pattern recognition.