Pattern Recognition 46 (2013) 1879–1890

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Learning dictionary on manifolds for image classification

Bao-Di Liu [a,*], Yu-Xiong Wang [a], Yu-Jin Zhang [a], Bin Shen [b]

[a] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[b] Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Available online 23 November 2012

Keywords:
Sparse coding
Image classification
Locally linear embedding
Coordinate descent
Manifold

ABSTRACT

At present, dictionary based models have been widely used in image classification. The image features are approximated as a linear combination of bases selected from the dictionary in a sparse space, resulting in compact patterns. The features applied to image classification usually reside on low dimensional manifolds embedded in a high dimensional ambient space; traditional sparse coding algorithm, however, does not consider this topological structure. It can be characterized naturally by linear coefficients that reconstruct each data point from its neighbors. One of the central issues here is how to determine the neighbors and learn the coefficients. In this paper, the geometrical structures are encoded in two situations. In simple cases when data points distribute on a single manifold, it is explicitly modeled by locally linear embedding algorithm combined with k-nearest neighbors. Nevertheless, in real-world scenarios, complex data points often lie on multiple manifolds. Sparse representation algorithm combined with k-nearest neighbors is instead utilized to construct the topological structures, because it is capable of approximating the data point by selecting its homogenous neighbors adaptively to guarantee the smoothness of each manifold. After obtaining the local fitting relationship, these two topological structures are then embedded into sparse coding algorithm as regularization terms to formulate the corresponding objective functions of dictionary learning on single manifold (DLSM) and dictionary learning on multiple manifolds (DLMM), respectively. Upon this, a coordinate descent scheme is proposed to solve the unified optimization problems. Experimental results on several benchmark data sets, such as Caltech-256, Caltech-101, Scene 15, and UIUC-Sports, show that our proposed algorithms equal or outperform other state-of-the-art image classification algorithms.

1. Introduction

Recently, image classification, which aims at associating images with semantic labels automatically, has become quite a significant topic in computer vision. The most common framework is the discriminative model [1,2,3,4]. Typically, learning vocabulary by applying k-means clustering on image patches (also called bag-of-word model) [5] and combining with hard-assignment vector quantization (VQ) is the most popular method at the early stage. Hard-assignment VQ method treats an image as a collection of "Visual words" (vocabulary), and each image patch is mapped to one word in the vocabulary. After that, several variants of vocabulary combined with vector quantization have been proposed in improving the classification performance recently. Jurie and Triggs [6] considered the densest point rather than the center of uniform region as the clustering center to learn vocabulary. Lazebnik et al. [1] extended the hard-assignment VQ method with spatial pyramid matching (SPM) kernel to compensate the loss of spatial information. Wu and Rehg [3] proposed histogram intersection kernel to form the vocabulary and applied one-class SVM to retrain the vocabulary and achieved satisfying results. In the year 2009, Yang et al. [2] used sparse coding algorithm for learning dictionary and coding images, resulting in state-of-the-art performance in image classification. Compared with hard-assignment VQ method, sparse coding algorithm can achieve sparse approximations with lower reconstruction.

On the other hand, some recent research work suggested that image space is actually a smooth low dimensional sub-manifold embedded in a high dimensional ambient space. Many manifold learning algorithms, such as locally linear embedding(LLE) [7], ISOMAP [8], and Laplacian Eigenmaps [9] were proposed to explicitly explore the intrinsic topological structure, which could significantly enhance the dimensionality reduction performance. All these algorithms consider the locality property and preserve it when learning the patterns. As suggested in [10], locality was more essential than sparsity, since locality can lead to sparsity while sparsity cannot cause locality. Therefore, more and more researchers focused on locality-preserving during dictionary learning for image classification. van Gemert et al. [11] proposed kernel codebook and

* Corresponding author. Tel.: +86 10 62798336; fax: +86 10 62770317.
E-mail addresses: lbd08@mails.tsinghua.edu.cn (B.-D. Liu), albertwyx@gmail.com (Y.-X. Wang), zhang-yj@mail.tsinghua.edu.cn (Y.-J. Zhang), stanshenbin@gmail.com (B. Shen).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.11.018

soft-assignment vector quantization to preserve the local properties. Wang et al. [12] considered that each word in the vocabulary was on a manifold, and utilized locally linear coding [7] for vector quantization to preserve the local information on vocabulary. Yan et al. [13] proposed a general framework for dimensionality reduction called graph embedding and claimed that most of graph embedding methods could be unified within this framework. Gao et al. [14] proposed to incorporate the histogram intersection kernel based Laplacian matrix into the objective function of sparse coding to enforce the consistence in sparse representation of similar local features. Yang et al. [15] proposed intrinsic graph and penalty graph to preserve the intrinsic graph and laid off the penalty graph in a supervised way. Zheng et al. [16] proposed to incorporate the vector quantization based Laplacian matrix into the objective function of sparse coding. Lu and Peng [17] proposed to incorporate the hypergraph (vertex, hyperedge, incidence matrix and hyperedge weights) regularization term into the objective function of sparse coding. Ramamurthy et al. [18] assumed that data points distributed on the same manifold and proposed a manifold projection to improve traditional sparse coding. Shen and Si [19] proposed to construct multiple manifolds structures by sparse representation algorithm, however, locality was not explicitly considered. Liu et al. [20] proposed a discriminant sparse coding scheme to incorporate the label information into sparse coding algorithm.

Inspired by both the superior performance of sparse coding based dictionary learning for image classification and the enhancement of dimension reduction on manifold, dictionary learning on single manifold (DLSM) and multiple manifolds (DLMM) are integrated and proposed in this paper. For DLSM, the intrinsic topological structure of the original data is explicitly modeled under the assumption that the original data could be fit by the linear combination of its all k-nearest neighbors which is then embedded into the objective function of sparse coding algorithm as the regularization term, while for DLMM, the original data could be fit by the linear combination of its neighbors, which only lie on the same manifold with the original data, selected from k-nearest neighbors. After that, a coordinate descent scheme [21] with guaranteed convergence is proposed to solve the unified optimization problems. The proposed dictionary learning methods on manifolds for image classification are evaluated on four benchmark data sets and achieve higher classification accuracy than traditional sparse coding algorithm.

The major contributions of this paper are as follows. First, two types of topological structures are proposed and analyzed on why and how these two graph models construct the manifold structures. Second, a coordinate descent scheme is proposed to solve the DLSM and DLMM as unified optimization problems. Third, the proposed algorithms are evaluated in image classification task; as shown experimentally in Section 6, the performances of our algorithms equal or outperform other state-of-the-art image classification algorithms on several benchmark data sets.

The rest of the paper is organized as follows. Section 2 overviews some related work contributing to image classification. Dictionary learning methods on manifolds are proposed in Section 3. The solution to the minimization of the objective function and guaranteed convergence are elaborated in Section 4. And implementation for image coding and spatial pooling is given in Section 5. Then, experimental results and analysis are shown in Section 6. Finally, discussions and conclusions are drawn in Section 7.

2. Related work

In this section, some abbreviations and notations are given first. Then several popular dictionary learning and image coding methods are listed.

Table 1
Some abbreviations.

Abbreviations	Full name
DLMM	Dictionary learning on multiple manifolds
DLSM	Dictionary learning on single manifold
HIK	Histogram intersection kernel
HIKVQ	VQ method combining with HIK
KC	Kernel codebook
KScSPM	Kernel ScSPM
KSPM	Nonlinear kernel SPM
LLC	Locality-constrained linear coding
LLE	Locally linear embedding
LScSPM	Laplacian ScSPM
OCSVM	One-class SVM for generating codebook
SC	Sparse coding
ScSPM	Sparse coding based linear SPM
SPM	Spatial pyramid matching
VQ	Vector quantization

2.1. Some abbreviations and notations

For convenience, some abbreviations are as shown in Table 1.

Let $\boldsymbol{X}\in\mathbb{R}^{D\times N}$ represent the local descriptors extracted from training images for learning dictionary, where D is the dimension of $\boldsymbol{X}$, and N is the number of samples in $\boldsymbol{X}$. Let $\boldsymbol{B}\in\mathbb{R}^{D\times K}$ be the dictionary, and $\boldsymbol{S}\in\mathbb{R}^{K\times N}$ be the corresponding codes, where K is the size of the dictionary. Let $\boldsymbol{Y}\in\mathbb{R}^{D\times M}$ be the features extracted from an image $\boldsymbol{J}$, and $\boldsymbol{V}\in\mathbb{R}^{K\times M}$ represent the corresponding codes. Let $\boldsymbol{Z}\in\mathbb{R}^{(K\times L)\times 1}$ be a vector after spatial pooling to represent the image, where L is the total number of regions in each layer split by SPM ($L=1+4+16=21$). Let $\|\cdot\|_F^2$ represent the Frobenius norm. Let $\boldsymbol{A}_{\bullet n}$ and $\boldsymbol{A}_{m\bullet}$ denote the nth column and mth row vectors of matrix $\boldsymbol{A}$, respectively. Let $\mathrm{tr}\{\boldsymbol{A}\}$ represent the trace of matrix $\boldsymbol{A}$. Let knn represent the number of neighbors for each descriptor.

2.2. Several popular dictionary learning and image coding methods

The aim of dictionary learning [22] is to find the optimum dictionary to make $\boldsymbol{X}\approx\boldsymbol{BS}$. After obtaining the dictionary, for each image $\boldsymbol{J}$, we assume that f_c and f_p denote coding and pooling operators. The image coding and pooling step can be formulated as

$$\boldsymbol{V}=f_c(\boldsymbol{Y}),\quad \boldsymbol{Z}=f_p(\boldsymbol{V}) \tag{1}$$

Usually, image coding is related to dictionary learning, and pooling strategy is max [2] or average. LIBSVM [23] is adopted for classifier training. Therein, dictionary learning and image coding are the most focused steps. Several popular dictionary learning and image coding methods are as follows.

Hard-assignment vector quantization for image classification: Hard-assignment vector quantization method is the most popular method, which solves the following optimization problem:

$$\min_{\boldsymbol{B},\boldsymbol{S}}\quad f(\boldsymbol{B},\boldsymbol{S})=\|\boldsymbol{X}-\boldsymbol{BS}\|_F^2$$
$$\text{s.t.}\quad \|\boldsymbol{S}_{\bullet i}\|_0=1,\ \|\boldsymbol{S}_{\bullet i}\|_1=1,\ \boldsymbol{S}_{\bullet i}\geq 0,\quad \forall i \tag{2}$$

The above optimization problem is usually solved by k-means algorithm. $\|\boldsymbol{S}_{\bullet i}\|_0=1$ means that there is only one nonzero element in each $\boldsymbol{S}_{\bullet i}$, i.e. each $\boldsymbol{X}_{\bullet i}$ is represented by only one basis in dictionary $\boldsymbol{B}$. $\|\boldsymbol{S}_{\bullet i}\|_1=1$ and $\boldsymbol{S}_{\bullet i}\geq 0$ mean that the weight for the corresponding basis is 1 to represent $\boldsymbol{X}_{\bullet i}$.

The distance between local feature and dictionary (also called codebook) utilizes histogram intersection kernel in [3]. The objective function and kernel function are defined as (3) and (4),

respectively

$$\min_{\boldsymbol{B},\boldsymbol{S}} \quad f(\boldsymbol{B},\boldsymbol{S}) = \|\phi(\boldsymbol{X}) - \phi(\boldsymbol{B})\boldsymbol{S}\|_F^2$$
$$\text{s.t.} \quad \|\boldsymbol{S}_{\bullet i}\|_0 = 1,\ \|\boldsymbol{S}_{\bullet i}\|_1 = 1,\ \boldsymbol{S}_{\bullet i} \geq 0, \quad \forall i \tag{3}$$

$$\kappa_{hik}(\boldsymbol{X}_{\bullet i},\boldsymbol{B}_{\bullet j}) = \phi(\boldsymbol{X}_{\bullet i})^T\phi(\boldsymbol{B}_{\bullet j}) = \sum_{d=1}^{D} \min(\boldsymbol{X}_{di},\boldsymbol{B}_{dj}) \tag{4}$$

Hard-assignment vector quantization method is simple and intuitive. However, the reconstruction error would be too high due to too much constraints, thus leading to lower performance. Notably, Hard-assignment vector quantization obtains better performance with histogram intersection kernel than that with linear kernel.

Soft-assignment vector quantization for image classification: The soft-assignment vector quantization method also adopts the optimization problem in Eq. (2) to build dictionary. To reduce the reconstruction error, soft-assignment vector quantization method loosens the restrictions of descriptor's coding, which uses a linear combination of multiple bases to approximate $\boldsymbol{X}_{\bullet i}$

$$\boldsymbol{S}_{ki} = \frac{\kappa(\boldsymbol{X}_{\bullet i},\boldsymbol{B}_{\bullet k})}{\sum_{m=1}^{K} \kappa(\boldsymbol{X}_{\bullet i},\boldsymbol{B}_{\bullet m})} \tag{5}$$

The kernel function κ here can be any types of kernels, [11] adopted gaussian kernel

$$\kappa_{gauss}(\boldsymbol{X}_{\bullet i},\boldsymbol{B}_{\bullet j}) = \exp(-\beta\|\boldsymbol{X}_{\bullet i} - \boldsymbol{B}_{\bullet j}\|_2^2) \tag{6}$$

where β is the soft assignments factor. The soft-assignment vector quantization achieves good performance because the Gaussian kernel is performed as a local constraint, i.e. each bin of $\boldsymbol{S}_{\bullet i}$ is corresponding to the weight of bases, and the smaller distance between descriptor $\boldsymbol{X}_{\bullet i}$ and dictionary $\boldsymbol{B}$ will get a higher weight.

Locality-constrained linear coding (LLC) for image classification: Locality-constrained linear coding [12] for image classification uses the following criteria:

$$\min_{\boldsymbol{B},\boldsymbol{S}} \quad f(\boldsymbol{B},\boldsymbol{S}) = \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{S}\|_F^2 + \lambda\|\kappa_{gauss}(\boldsymbol{X},\boldsymbol{B}) \odot \boldsymbol{S}\|_F^2$$
$$\text{s.t.} \quad \boldsymbol{1}^T\boldsymbol{B}_{\bullet k} = 1, \quad \forall k \tag{7}$$

where

$$\kappa_{gauss}(\boldsymbol{X},\boldsymbol{B}) = [\varkappa_{gauss}(\boldsymbol{X}_{\bullet 1},\boldsymbol{B}),\ldots,\varkappa_{gauss}(\boldsymbol{X}_{\bullet N},\boldsymbol{B})]$$

$$\varkappa_{gauss}(\boldsymbol{X}_{\bullet i},\boldsymbol{B}) = [\exp(dist(\boldsymbol{X}_{\bullet i},\boldsymbol{B}_{\bullet 1})/\sigma),\ldots,\exp(dist(\boldsymbol{X}_{\bullet i},\boldsymbol{B}_{\bullet K})/\sigma)]^T$$

$\odot$ denotes the element-wise multiplication, λ is used for adjusting the locality. The larger the λ is, the more locality is considered. LLC has several advantages. First, it can better reconstruct the raw descriptors than VQ method. Second, the raw descriptor and its neighboring bases construct a local coordinate system, which would lead to local smooth sparsity, i.e. the explicit locality adopted in LLC ensures that similar patches have similar codes. Third, this optimization problem has the analytical solution. The LLC method obtains better performance than VQ-based method.

Sparse coding for image classification: Sparse coding (SC) can be considered as methods of rearranging the structure of the original data in order to make the energy compact under over-complete or non-orthogonal bases. Hence, the data point can be represented as a linear combination of only few active bases that possess overwhelmingly majority energy of the data. Sparse coding method was firstly introduced into image classification in [2]. Unlike VQ method, sparse coding loosens the constraint on the codes, each code can be represented by the linear combination of several bases to minimize the reconstruction error. The sparse coding algorithm can be written as the following formula:

$$\min_{\boldsymbol{B},\boldsymbol{S}} \quad f(\boldsymbol{B},\boldsymbol{S}) = \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1$$
$$\text{s.t.} \quad \|\boldsymbol{B}_{\bullet i}\|_2 = 1, \quad \forall i = 1,2,\ldots,K \tag{8}$$

The regularization term is to control sparsity in $\boldsymbol{S}$, where α is a regularization parameter to control the tradeoff between fitting goodness and sparseness. Sparse coding method has several advantages. First, it can learn over-complete bases, i.e. $K > D$. That is to say, each feature has opportunity to choose better base. Second, it can achieve less reconstruction error than VQ method and LLC method. Third, it can capture salient patterns of local features. So image classification by sparse coding framework achieves state-of-the-art performance on several benchmarks such as Caltech-101, Scene 15, etc.

Locality-constrained sparse coding for image classification: Sparse coding for image classification shows good performance, however, due to the over-complete dictionary and the independent coding process [14], the locality or the geometrical structure among the instances to be encoded are lost. Gao et al. [14] and Zheng et al. [16] proposed locality-constrained sparse coding to preserve the local manifold structure of the instances by embedding the Laplacian matrix $\boldsymbol{G}$ into sparse coding algorithm. The objective function for locality-constrained sparse coding is as follows:

$$\min_{\boldsymbol{B},\boldsymbol{S}} \quad f(\boldsymbol{B},\boldsymbol{S}) = \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1 + \eta\ \mathrm{tr}\{\boldsymbol{S}\boldsymbol{G}\boldsymbol{S}^T\}$$
$$\text{s.t.} \quad \|\boldsymbol{B}_{\bullet i}\|_2 = 1, \quad \forall i = 1,2,\ldots,K \tag{9}$$

where η is the regularization parameter balancing the weight between the fitting goodness and locality preservation. The Laplacian matrix $\boldsymbol{G} \in \mathbb{R}^{N\times N}$ can be obtained by $\boldsymbol{G} = \boldsymbol{W} - \boldsymbol{U}$, where [16] constructed the k-nearest neighbors' distance matrix $\boldsymbol{U}$ as follows:

$$\boldsymbol{U}_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{X}_{\bullet j} \in \mathcal{N}_{knn}\{\boldsymbol{X}_{\bullet i}\} \\ 0 & \text{if } \boldsymbol{X}_{\bullet j} \notin \mathcal{N}_{knn}\{\boldsymbol{X}_{\bullet i}\} \end{cases} \tag{10}$$

Gao et al. [14] constructed the k-nearest neighbors' distance matrix $\boldsymbol{U}$ as follows,

$$\boldsymbol{U}_{ij} = \begin{cases} \sum_{d=1}^{D} \min(\boldsymbol{X}_{di},\boldsymbol{X}_{dj}) & \text{if } \boldsymbol{X}_{\bullet j} \in \mathcal{N}_{knn}\{\boldsymbol{X}_{\bullet i}\} \\ 0 & \text{if } \boldsymbol{X}_{\bullet j} \notin \mathcal{N}_{knn}\{\boldsymbol{X}_{\bullet i}\} \end{cases} \tag{11}$$

$\mathcal{N}_{knn}\{\boldsymbol{X}_{\bullet i}\}$ represents k-nearest neighbors of $\boldsymbol{X}_{\bullet i}$. The degree matrix $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ is a diagonal matrix and $\boldsymbol{W}_{ii} = \sum_{j=1}^{N} \boldsymbol{U}_{ij}$.

3. Proposed dictionary learning methods on manifolds

In this section, dictionary learning methods on manifolds are proposed for image classification. First, the geometrical structure of the single manifold is constructed by locally linear embedding algorithm combined with k-nearest neighbors. Then, sparse representation combined with k-nearest neighbors is adopted to construct the geometrical structure of the multiple manifolds. After that, we preserve these two structures to form dictionary learning on single manifold (DLSM) method and dictionary learning on multiple manifolds (DLMM) method, respectively.

3.1. Image features lie on a single manifold

In the classification application, the image data are often sampled from a nonlinear low dimensional manifold embedding in a high dimensional space (see in Fig. 1 (a)). In the situation of a single manifold, each data point and its neighboring points are close to or event lie on a locally flat patch of the manifold. A very straightforward idea in modeling this global nonlinear structure is to learn from locally linear fitting.

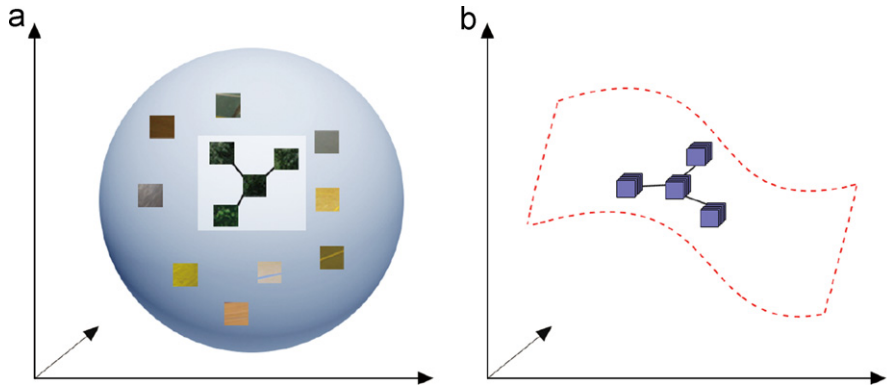

Fig. 1. Data in the features space and codes space. (a) The original image features lying on the manifold and (b) the corresponding feature codes.

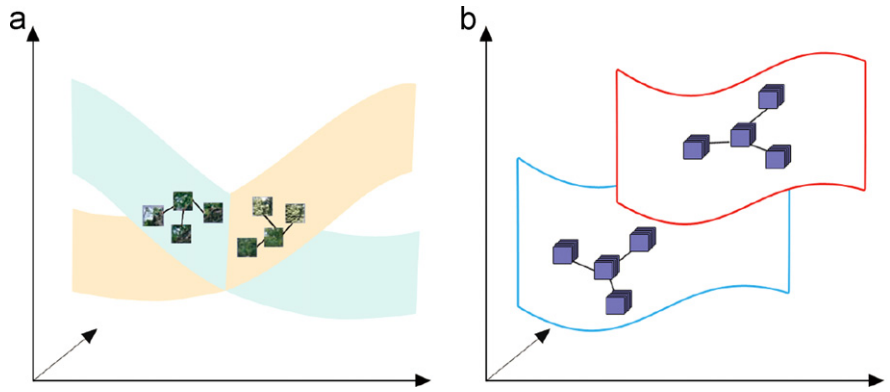

Fig. 2. Data in the features space and codes space. (a) The original image features lying on the two manifolds and (b) the corresponding feature codes.

Let $\boldsymbol{X}_{\bullet i}$ represent the ith data point, and $\mathcal{N}_{knn}\{\boldsymbol{X}_{\bullet i}\}$ represent its k-nearest neighbors. For most cases, five nearest neighbors are considered. The local topology around $\boldsymbol{X}_{\bullet i}$ can be characterized by the linear combination of its neighbors as follows:

$$\boldsymbol{X}_{\bullet i} \approx \boldsymbol{X}\boldsymbol{U}_{i\bullet}$$
$$\text{s.t.} \sum_{j=1}^{N} \boldsymbol{U}_{ij} = 1 \& \boldsymbol{U}_{ij} = 0 \text{ for } \boldsymbol{X}_{\bullet j} \notin \mathcal{N}_{knn}\{\boldsymbol{X}_{\bullet i}\} \tag{12}$$

where $\boldsymbol{U} \in \mathbb{R}^{N\times N}$ is the coefficient matrix.

In this paper, k is set to 5 and locally linear embedding (LLE) [7] algorithm is used to obtain the coefficient matrix $\boldsymbol{U}$ of the single manifold's graph.

3.2. Image features lie on multiple manifolds

The majority of real-world data often reside on multiple manifolds [24], whose manifold structures, such as curvature variation, may differ a lot. They may also overlap or intersect (Fig. 2(a)). This makes the whole scenario much more complicated. By manifold learning, we hope to analyze these manifolds separately. We still model the underlying geometrical structure patchwise. Thus for a certain data point, the true neighborhood on the same manifold instead of the entire space is needed firstly. However, existing manifold learning algorithms, such as locally linear embedding algorithm and Laplacian eigenmaps, define the neighborhood relationship only according to the Euclidean distance, which is insufficient. In other words, data points on a nearby manifold may be involved mistakenly as the neighboring point in the fix-sized neighborhood used in LLE. To preserve the topology of multiple manifolds, we assume that each manifold is smooth [25], and the local data points lying on the same manifold are homogenous in a certain sense such as direction, while overlaps or intersections under different manifolds are not ruled out. We propose to utilize sparse representation algorithm combined with k-nearest neighbors to fit the topology of these multiple manifolds, with the consideration that sparse representation is capable of approximating the data point by selecting the homogenous neighbors adaptively and thus guarantees the smoothness of the manifold. This is helpful in identifying the neighborhood relationship on the same single manifold. And in fact we want to preserve the structure of the same single manifold in the following processing.

To demonstrate our conclusion, we carry out an experiment on ORL face database. The ORL face database contains 10 images for each of forty human faces with each lying on the same manifold. Each data point and its k-nearest neighbors (5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 100, 150, 200, 250, 300, 350, 399) are executed twice by locally linear embedding algorithm and sparse representation algorithm, respectively. For each data point, given the initial size of neighborhood (characterized by the number of selected neighboring points), the percentages of number and absolute coefficients summation of neighbors on the same manifold (belongs to the same human) are calculated. And the means of the percentages of number and absolute coefficients summation of neighbors lying on the same manifold for all data points are given. For locally linear embedding, the initial nearby points are all used as the neighboring points; whereas, for sparse representation algorithm, the active points with nonzero coefficients after sparse coding, the subset of the initial points, are selected as the neighboring points. Fig. 3 shows the percentage of number of neighbors lying on the same manifold with varying number of neighbors. From Fig. 3, we can see that most data points and their five neighbors lie on the same manifold. The percentage of number of neighbors on the same manifold is 83.45% among k-nearest neighbors, and 83.33% selected after

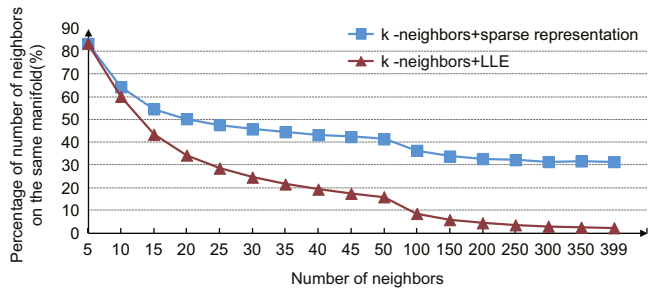

Fig. 3. The percentage of neighbors on the same manifold with varying number of neighbors. The percentage is the ratio of the number of the neighbors lying on the same manifold with nonzero fitting coefficients to the number of the overall neighbors with nonzero fitting coefficients.

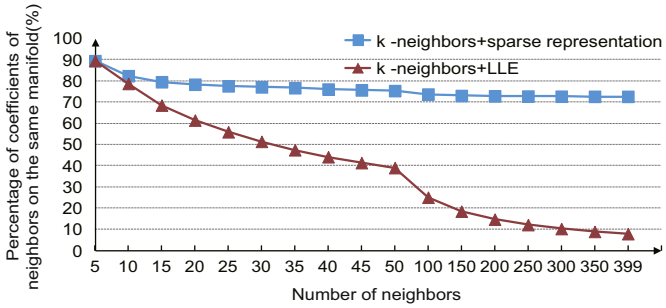

Fig. 4. The percentage of summation of absolute fitting coefficients of neighbors on the same manifolds with varying number of neighbors. The percentage is the ratio of the sum of absolute nonzero fitting coefficients of neighbors lying on the same manifold to the sum of absolute nonzero fitting coefficients of neighbors.

sparse representation combined with k-nearest neighbors. With the increasing number of neighbors, this value decreases till 2.26% for 399 among k-nearest neighbors. This indicates that almost all data points' neighbors are not on the same manifold. However, this percentage achieves 31.37% selected after sparse representation. So sparse representation algorithm combined with k-nearest neighbors is capable of selecting effective ones from neighbors and removing the neighbors that do not lie on the same manifold especially when the initial nearby points distribute on the different manifold structures. Another statistical quantity, the percentage of summation of absolute fitting coefficients of neighbors on the same manifold in all the coefficients constructing the graph model, is much more convincing than the percentage of number of neighbors on the same manifold. From Fig. 4, we can see that even for 399 neighbors, (i.e. only nine neighbors are on the same manifold, and other 390 neighbors distribute on the other 39 manifolds), the percentage of summation of absolute fitting coefficients of neighbors on the same manifold by sparse representation combined with k-nearest neighbors is 72.54%, much higher than 7.89% by locally linear embedding combined with k-nearest neighbors. And with the number of neighbors increasing, this percentage by sparse representation combined with k-nearest neighbors maintains at 72%. This result is beneficial for complex scenario with multiple manifolds. From the comparison, sparse representation combined with k-nearest neighbors is capable of capturing the similar response as neighbors on the same manifold and therefore constructing the graph model for multiple manifolds in an unsupervised way. In other words, it is sort of resistant to the initial k-nearest neighboring points. Here, we use $l1$-norm minimization technique:

$$f(\boldsymbol{U}_{i\bullet}^{knn}) = \|\boldsymbol{X}_{\bullet i} - \mathcal{N}_{knn}(\boldsymbol{X}_{\bullet i})\boldsymbol{U}_{i\bullet}^{knn}T\|^2 + 2\lambda|\boldsymbol{U}_{i\bullet}^{knn}| \tag{13}$$

where $\boldsymbol{X}_{\bullet i} \in \mathbb{R}^{D\times 1}$ represents the ith data point of $\boldsymbol{X}$. $\mathcal{N}_{knn}(\boldsymbol{X}_{\bullet i}) \in \mathbb{R}^{D\times knn}$ represents the data point's k-neighbors. $\boldsymbol{U}_{i\bullet}^{knn} \in \mathbb{R}^{1\times knn}$ is the corresponding sparse coefficient under the bases $\mathcal{N}_k(\boldsymbol{X}_{\bullet i})$.

$$\boldsymbol{U}_{ij} = \begin{cases} \boldsymbol{U}_{ij}^{knn} & \text{if } \boldsymbol{X}_{\bullet j} \in \mathcal{N}_{knn}(\boldsymbol{X}_{\bullet i}) \\ 0 & \text{if } \boldsymbol{X}_{\bullet j} \notin \mathcal{N}_{knn}(\boldsymbol{X}_{\bullet i}) \end{cases} \tag{14}$$

In this section and Section 6.6.3, feature-sign search algorithm [22] is used, and the λ in Eq. (13) is set to 80. In the later part of the paper for image classification, knn is set to 10, feature-sign search algorithm [22] is adopted to obtain the coefficient matrix $\boldsymbol{U}$ of multiple manifolds graph, and λ is set to 0.05.

3.3. The objective function of dictionary learning on manifolds

The geometrical structure represented by $\boldsymbol{U} \in \mathbb{R}^{N\times N}$ is preserved during the sparse coding procedure. Specifically, when the features $\boldsymbol{X}$ are transformed to the new features $\boldsymbol{S} \in \mathbb{R}^{K\times N}$, the local geometrical structure is preserved through the relationship $\boldsymbol{U}$ for $\boldsymbol{S}$, which can be obtained by minimizing the following term (see in Figs. 1 and 2):

$$\sum_i \|\boldsymbol{S}_{\bullet i} - \boldsymbol{S}\boldsymbol{U}_{\bullet i}\|^2 = \|\boldsymbol{S} - \boldsymbol{S}\boldsymbol{U}\|_F^2 = \mathrm{tr}(\boldsymbol{S}(\boldsymbol{I}-\boldsymbol{U})(\boldsymbol{I}-\boldsymbol{U})^T\boldsymbol{S}^T) = \mathrm{tr}\{\boldsymbol{S}\boldsymbol{G}\boldsymbol{S}^T\} \tag{15}$$

where $\boldsymbol{G} \in \mathbb{R}^{N\times N}$, $\mathrm{tr}\{\boldsymbol{S}\boldsymbol{G}\boldsymbol{S}^T\}$ is the preservation of local geometry structure in the sparse space.

We incorporate the preservation term of local geometry structure $\mathrm{tr}\{\boldsymbol{S}\boldsymbol{G}\boldsymbol{S}^T\}$ into sparse coding algorithm as a regularization term. The dictionary learning on single manifold (DLSM) algorithm and dictionary learning on multiple manifolds (DLMM) can be unified into the same framework and gained from minimizing the following objective function:

$$\boldsymbol{O}_{DL} = \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1 + \eta\ \mathrm{tr}\{\boldsymbol{S}\boldsymbol{G}\boldsymbol{S}^T\} \tag{16}$$

where the first term is the fitting error, the second term encourages the sparseness, and the third term is the locality constraint. η is the regularization parameter balancing the weight between the fitting goodness and geometrical structure. α is the regularization parameter adjusting the sparsity. When applying the $\boldsymbol{U}$ in Eqs. (12)–(15), dictionary learning on single manifold (DLSM) can be obtained by solving Eq. (16). When applying the $\boldsymbol{U}$ in Eqs. (13)–(15), dictionary learning on multiple manifolds (DLMM) can be obtained by solving Eq. (16). It is important to note that the form of our objective function is similar to that of [14,16]; however, it is quite different as for representing the topology. For the graph model preservation in [14,16], the author proposed to use Laplacian matrix to preserve the distance between the data point and its neighbors. Our DLSM method utilizes the weight matrix obtained by LLE algorithm combined with k-nearest neighbors to preserve the data point's fitting coefficients. And our DLMM method adopts the weight matrix obtained by sparse representation algorithm combined with k-nearest neighbors to preserve the data point's fitting coefficients. Our DLMM method has the advantage of handling multiple manifolds scenario.

4. Optimization of the objective function

In this section, we focus on solving the minimization of the objective function proposed in (16). This optimization problem is not jointly convex in both $\boldsymbol{B}$ and $\boldsymbol{S}$, while it is separately convex in either $\boldsymbol{B}$ or $\boldsymbol{S}$ with $\boldsymbol{S}$ or $\boldsymbol{B}$ fixed. So it can be decoupled into the following two optimization subproblems which can be solved by alternating minimizations.

Finding the sparse codes is as follows:

$$\min f(\boldsymbol{S}) = \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1 + \eta\ \mathrm{tr}\{\boldsymbol{S}\boldsymbol{G}\boldsymbol{S}^T\} \tag{17}$$

Learning bases are as follows:

$$\min \quad f(\boldsymbol{B}) = \|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2$$
$$\text{s.t.} \quad \|\boldsymbol{B}_{\bullet i}\|_2 = 1, \quad \forall i = 1,2,\ldots,K \tag{18}$$

In the following subsection, a coordinate descent algorithm is introduced to resolve these two optimization problems.

4.1. Finding sparse codes

By fixing $\boldsymbol{B}$, we update $\boldsymbol{S}$ to decrease the value of the objective function

$$\begin{aligned} f(\boldsymbol{S}) &= \|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1 + \eta\ \mathrm{tr}\{\boldsymbol{S}\boldsymbol{G}\boldsymbol{S}^T\} \\ &= \mathrm{tr}\{\boldsymbol{X}^T\boldsymbol{X} - 2\boldsymbol{X}^T\boldsymbol{B}\boldsymbol{S} + \boldsymbol{S}^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{S}\} + 2\alpha\|\boldsymbol{S}\|_1 + \eta\ \mathrm{tr}\{\boldsymbol{S}\boldsymbol{G}\boldsymbol{S}^T\} \\ &= \mathrm{tr}\{\boldsymbol{X}^T\boldsymbol{X}\} - 2\sum_{n=1}^{N}[\boldsymbol{X}^T\boldsymbol{B}]_{n\bullet}\boldsymbol{S}_{\bullet n} + \sum_{n=1}^{N}\boldsymbol{S}_{\bullet n}^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{S}_{\bullet n} \\ &\quad + 2\alpha\sum_{k=1}^{K}\sum_{n=1}^{N}|\boldsymbol{S}_{kn}| + \eta\sum_{k=1}^{K}\boldsymbol{S}_{k\bullet}\boldsymbol{G}\boldsymbol{S}_{k\bullet}^T \end{aligned} \tag{19}$$

Ignoring the constant term $\mathrm{tr}\{\boldsymbol{X}^T\boldsymbol{X}\}$, the objective function of $\boldsymbol{S}_{kn}$ reduces to (20) with $\boldsymbol{B}$ and $\{\boldsymbol{S}_{1n},\boldsymbol{S}_{2n},\ldots,\boldsymbol{S}_{kn}\}/\boldsymbol{S}_{kn}$ fixed

$$\begin{aligned} f(\boldsymbol{S}_{kn}) &= \boldsymbol{S}_{kn}^2\{[\boldsymbol{B}^T\boldsymbol{B}]_{kk} + \eta\boldsymbol{G}_{nn}\} + 2\alpha|\boldsymbol{S}_{kn}| \\ &\quad + 2\boldsymbol{S}_{kn}\left\{\sum_{l=1,l\neq k}^{K}[\boldsymbol{B}^T\boldsymbol{B}]_{kl}\boldsymbol{S}_{ln} + \eta\sum_{r=1,r\neq n}^{N}\boldsymbol{G}_{nr}\boldsymbol{S}_{kr} - [\boldsymbol{B}^T\boldsymbol{X}]_{kn}\right\} \\ &= \boldsymbol{S}_{kn}^2\{[\boldsymbol{B}^T\boldsymbol{B}]_{kk} + \eta\boldsymbol{G}_{nn}\} + 2\alpha|\boldsymbol{S}_{kn}| - 2\boldsymbol{S}_{kn}\boldsymbol{H}_{kn} \end{aligned} \tag{20}$$

where $\boldsymbol{H}_{kn} = [\boldsymbol{B}^T\boldsymbol{X}]_{kn} - \sum_{l=1,l\neq k}^{K}[\boldsymbol{B}^T\boldsymbol{B}]_{kl}\boldsymbol{S}_{ln} - \eta\sum_{r=1,r\neq n}^{N}\boldsymbol{G}_{nr}\boldsymbol{S}_{kr}$.

Here, $[\boldsymbol{B}^T\boldsymbol{B}]_{kk} = 1$, and $\boldsymbol{G}_{nn} > 0$. So $f(\boldsymbol{S}_{kn})$ is piece-wise parabolic function that opens up. Based on the convexity and monotonic property of the parabolic function, it is not difficult to know that $f(\boldsymbol{S}_{kn})$ reaches the minimum at the unique point

$$\boldsymbol{S}_{kn} = \{\max\{\boldsymbol{H}_{kn},\alpha\} + \min\{\boldsymbol{H}_{kn},-\alpha\}\}/\{1+\eta\boldsymbol{G}_{nn}\} \tag{21}$$

4.2. Learning bases

Without the sparseness regularization term in (17) and additional constraints in (18), $\boldsymbol{S}_{k\bullet}$ and $\boldsymbol{B}_{\bullet k}$ are dual in objective function $\|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2$ for $\forall k \in \{1,2,\ldots,K\}$. Hence, $\forall d \in \{1,2,\ldots,D\}$, $k \in \{1,2,\ldots,K\}$, with $\{\boldsymbol{B}_{pq,p=1,2,\ldots,D,q=1,2,\ldots,K}\}/\boldsymbol{B}_{dk}$ and $\boldsymbol{S}$ fixed, the unconstrained single variable minimization problem of (18) has the closed-form solution

$$\boldsymbol{B}_{dk} = \arg\min_{\boldsymbol{B}_{dk}}\|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2 = \frac{[\boldsymbol{X}\boldsymbol{S}^T]_{dk} - \sum_{l=1,l\neq k}^{K}\boldsymbol{B}_{dl}[\boldsymbol{S}\boldsymbol{S}^T]_{lk}}{[\boldsymbol{S}\boldsymbol{S}^T]_{kk}} \tag{22}$$

while $\|\boldsymbol{S}_{k\bullet}\|_1 > 0$.

Since the optimal value for $\boldsymbol{B}_{dk}$ does not depend on the other entries in the same column, the objective function of $\boldsymbol{B}_{\bullet k}$ reduces to (23) with $\boldsymbol{S}$ fixed

$$f(\boldsymbol{B}_{\bullet k}) = [\boldsymbol{S}_{k\bullet}[\boldsymbol{S}_{k\bullet}]^T][[\boldsymbol{B}_{\bullet k}]^T\boldsymbol{B}_{\bullet k}] + 2[\boldsymbol{B}_{\bullet k}]^T\{\tilde{\boldsymbol{B}}^k\boldsymbol{S}[\boldsymbol{S}_{k\bullet}]^T - \boldsymbol{X}[\boldsymbol{S}_{k\bullet}]^T\} \tag{23}$$

where

$$\tilde{\boldsymbol{B}}^k = \begin{cases} \boldsymbol{B}_{\bullet p}, & p \neq k \\ \boldsymbol{0}, & p = k \end{cases}.$$

When imposing the norm constraint, i.e. $\|\boldsymbol{B}_{\bullet k}\|_2 = [\boldsymbol{B}_{\bullet k}]^T\boldsymbol{B}_{\bullet k} = 1$, (23) becomes (24)

$$f(\boldsymbol{B}_{\bullet k}) = 2[\boldsymbol{B}_{\bullet k}]^T\{\tilde{\boldsymbol{B}}^k\boldsymbol{S}[\boldsymbol{S}_{k\bullet}]^T - \boldsymbol{X}[\boldsymbol{S}_{k\bullet}]^T\} + \boldsymbol{S}_{k\bullet}[\boldsymbol{S}_{k\bullet}]^T \tag{24}$$

Hence, the original constrained minimization problem becomes a linear programming under a unit norm constraint, whose solution is as follows:

$$\boldsymbol{B}_{\bullet k} = \frac{\boldsymbol{X}[\boldsymbol{S}_{k\bullet}]^T - \tilde{\boldsymbol{B}}^k\boldsymbol{S}[\boldsymbol{S}_{k\bullet}]^T}{\|\boldsymbol{X}[\boldsymbol{S}_{k\bullet}]^T - \tilde{\boldsymbol{B}}^k\boldsymbol{S}[\boldsymbol{S}_{k\bullet}]^T\|_2} \tag{25}$$

4.3. Convergence analysis

Assuming that $(\boldsymbol{B}^t,\boldsymbol{S}^t)$ is the result after the tth iteration, and $(\boldsymbol{B}^{t+1},\boldsymbol{S}^{t+1})$ is the result after the $(t+1)$th iteration. Since the exact minimum point is obtained by (21) and (25), each update operation will monotonically decrease the value of corresponding objective function. Considering that the objective function is obviously bounded below, and $f(\boldsymbol{B}^t,\boldsymbol{S}^t) \geq f(\boldsymbol{B}^t,\boldsymbol{S}^{t+1}) \geq f(\boldsymbol{B}^{t+1},\boldsymbol{S}^{t+1})$, it converges.

4.4. Overall algorithm

Our algorithm for learning dictionary is shown in Algorithm 1.

Algorithm 1. Learning dictionary on manifolds.

Require Data matrix $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ and K

$\boldsymbol{B} \leftarrow rand(D,K), \boldsymbol{B}_{\bullet k} = \frac{\boldsymbol{B}_{\bullet k}}{\|\boldsymbol{B}_{\bullet k}\|_2}\forall k$, $\boldsymbol{S} \leftarrow zeros(K,N)$
$iter = 0$
while $(f(iter)-f(iter+1))/f(iter) > 1e{-}5$ **do**
  $iter \leftarrow iter+1$
  Update $\boldsymbol{S}$:
  Compute $\boldsymbol{A} = (\boldsymbol{B}^T\boldsymbol{B}) \odot (\boldsymbol{1}-\boldsymbol{I})$, $\boldsymbol{C} = \boldsymbol{G} \odot (\boldsymbol{1}-\boldsymbol{I})$ and $\boldsymbol{E} = \boldsymbol{B}^T\boldsymbol{X}$
  for $n = 1; n \leq N; n{+}{+}$ **do**
    for $k = 1; k \leq K; k{+}{+}$ **do**
      $\boldsymbol{S}_{kn} = \{\max\{\boldsymbol{E}_{kn} - \boldsymbol{A}_{k\bullet}\boldsymbol{S}_{\bullet n} - \eta\boldsymbol{S}_{k\bullet}\boldsymbol{C}_{\bullet n}, \alpha\} + \min\{\boldsymbol{E}_{kn} - \boldsymbol{A}_{k\bullet}\boldsymbol{S}_{\bullet n} - \eta\boldsymbol{S}_{k\bullet}\boldsymbol{C}_{\bullet n}, -\alpha\}\}/(1+\eta\boldsymbol{G}_{nn})$
    end for
  end for
  Update $\boldsymbol{B}$:
  Compute $\boldsymbol{F} = (\boldsymbol{S}\boldsymbol{S}^T) \odot (\boldsymbol{1}-\boldsymbol{I})$, $\boldsymbol{W} = \boldsymbol{X}\boldsymbol{S}^T$
  for $k = 1; k \leq K; k{+}{+}$ **do**
    $\boldsymbol{B}_{\bullet k} = \frac{\boldsymbol{W}_{\bullet k} - \boldsymbol{B}\boldsymbol{F}_{\bullet k}}{\|\boldsymbol{W}_{\bullet k} - \boldsymbol{B}\boldsymbol{F}_{\bullet k}\|_2}$
  end for
  Update the objective function:
  $f = \|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1 + \eta\ \mathrm{tr}\{\boldsymbol{S}\boldsymbol{G}\boldsymbol{S}^T\}$
end while
return $\boldsymbol{B}$, and $\boldsymbol{S}$

Here, $\boldsymbol{1} \in \mathbb{R}^{K\times K}$ is a square matrix with all elements 1, $\boldsymbol{I} \in \mathbb{R}^{K\times K}$ is the identity matrix, and $\odot$ indicates element dot product. By iterating $\boldsymbol{S}$ and $\boldsymbol{B}$ alternately, the sparse codes are obtained, and the corresponding bases are learned.

5. Implementation for image coding and spatial pooling

After learning the dictionary $\boldsymbol{B}$, we implement our algorithm on image coding and spatial pooling. First, we find the relationship between the features $\boldsymbol{Y} \in \mathbb{R}^{D\times M}$ in image $\boldsymbol{J}$ and $\boldsymbol{X}$ (features for learning dictionary), let $\boldsymbol{P} \in \mathbb{R}^{N\times M}$ represent this relationship. $\boldsymbol{P}$ can be obtained according LLE algorithm [7] for DLSM and feature-sign search algorithm [22] for DLMM. Then, the objective function in the image coding step with dictionary fixed can be formulated as

$$\min f(\boldsymbol{V}) = \|\boldsymbol{Y}-\boldsymbol{B}\boldsymbol{V}\|_F^2 + 2\alpha\|\boldsymbol{V}\|_1 + \eta\|\boldsymbol{V}-\boldsymbol{S}\boldsymbol{P}\|_F^2 \tag{26}$$

$\boldsymbol{V}$ is the sparse codes for image features $\boldsymbol{Y}$. Similar to solve (17), the solution to the minimization of (26) is as follows:

$$\boldsymbol{V}_{km} = (\max\{\boldsymbol{H}_{km}, \alpha\} + \min\{\boldsymbol{H}_{km}, -\alpha\})/(1+\eta) \tag{27}$$

where $\boldsymbol{H}_{km} = [\boldsymbol{B}^T\boldsymbol{Y}]_{km} + \eta[\boldsymbol{SP}]_{km} - \sum_{l=1, l\neq k}^{K}[\boldsymbol{B}^T\boldsymbol{B}]_{kl}\boldsymbol{V}_{lm}$.

After coding each image, we split it into three levels (i.e. a three level spatial pyramid) with $L(1+4+16)$ regions. Then each region is pooled to a vector, and all L vectors are concatenated together to form the representation of the target image. The pooling strategy adopted is max. An image can be represented as follows:

$$\boldsymbol{Z} = \{\boldsymbol{hist}_1^{max}T, \boldsymbol{hist}_2^{max}T, \ldots, \boldsymbol{hist}_L^{max}T\} \tag{28}$$

where $\boldsymbol{hist}_l^{\max} = \max(\{\boldsymbol{V}_{\bullet i}\}_{i\in O^l})_{l=1,\ldots,L}$. O^l represents the features distributing in the lth region. Image coding and SPM pooling is shown in Algorithm 2.

Algorithm 2. Image coding and SPM pooling.

Require K, L, α, η, $\boldsymbol{Y}\in\mathbb{R}^{D\times M}$, $\boldsymbol{P}\in\mathbb{R}^{N\times M}$, $\boldsymbol{B}\in\mathbb{R}^{D\times K}$, $\boldsymbol{S}\in\mathbb{R}^{K\times N}$
$\boldsymbol{V}\leftarrow zeros(K,M)$, $\boldsymbol{Z}\leftarrow zeros(K\times L,1)$
Image Coding:
$iter=0$
while $(f(iter)-f(iter+1))/f(iter) > 1e-5$ **do**
  $iter\leftarrow iter+1$
  Update $\boldsymbol{V}$:
  Compute $\boldsymbol{A} = (\boldsymbol{B}^T\boldsymbol{B})\odot(\boldsymbol{1}-\boldsymbol{I})$ and $\boldsymbol{E} = \boldsymbol{B}^T\boldsymbol{Y} + \eta\boldsymbol{SP}$
  for $k=1; k\le K; k++$ **do**
    $\boldsymbol{V}_{k\bullet} = \{\max\{\boldsymbol{E}_{k\bullet} - \boldsymbol{A}_{k\bullet}\boldsymbol{V}, \alpha\} + \min\{\boldsymbol{E}_{k\bullet} - \boldsymbol{A}_{k\bullet}\boldsymbol{V}, -\alpha\}\}/(1+\eta)$
  end for
  Update the objective function:
  $f = \|\boldsymbol{Y}-\boldsymbol{BV}\|_F^2 + 2\alpha\|\boldsymbol{V}\|_1 + \eta\|\boldsymbol{Y}-\boldsymbol{SP}\|_F^2$
end while
SPM Pooling:
for $l=1; l\le L; l++$ **do**
  $\boldsymbol{hist}_l^{\max} = \max(\{V_{\bullet i}\}_{i\in O^l})$, $l=1,\ldots,L$
end for
$\boldsymbol{Z} = \{\boldsymbol{hist}_1^{max}T, \boldsymbol{hist}_2^{max}T, \ldots, \boldsymbol{hist}_L^{max}T\}$
return $\boldsymbol{Z}$

6. Experimental results

In this section, our DLSM and DLMM algorithms for image classification are evaluated on four benchmark data sets, including UIUC-Sports data set [26], Scene 15 data set [1,27,28], Caltech-101 data set [29] and Caltech-256 data set [30]. Parameter settings are given first. Then experimental results and analysis are demonstrated, and some discussions are listed finally.

6.1. Parameter settings

For each data set of image classification, the data are randomly split into the training set and the testing set based on published protocols. To make the results more convincing, the experimental process is repeated eight times, and the mean and standard deviation of the classification accuracy are recorded. Each image is resized with maximum side 300 pixels first.[1] As for the image features, densely sampling patches are extracted with the patch size and step size 16×16 and 8 pixels, respectively, and 128 dimensional SIFT descriptors [31] are obtained with grid size 4×4. The number of samples used for learning dictionary is about 120,000 and the dictionary size is 1024. Spatial pyramid matching kernel is embedded in the pooling step (the image is split into three layers, each of which has 1, 4, and 16 segments, respectively). The pooling strategy is max [2]. An image is represented as the concatenation of each segment with length 21,504 and normalized to 1 with $l2$-norm. Linear kernel SVM classifier and one against all multi-classification strategy are adopted, and LIBSVM [23] package is used.

There are two parameters: α and η. The parameter α is used for adjusting the sparsity of the codes. The bigger α is, the sparser the codes are. Yang et al. [2] have obtained the empirical value 0.15 for α. The parameter η is used for balancing the weight between the fitting goodness term and the geometrical structure preserving term. In this paper, α is set to 0.15 and η is set to 0.2 (for details, see Section 6.6.1).

For experiment of comparison on strong manifolds structures, the procedure is similar to the image classification in addition to the lack of pooling step. The dimension of features is $112*92 = 10{,}304$, and the dictionary size is 200. The parameter α is set to 0.15 and η is set to 0.8. The first five faces of each class are selected as the training set and the rest are the testing set. Linear kernel SVM classifier and one against all multi-classification strategy are adopted, and we use LIBSVM [23] package for the implementation.

6.2. UIUC-Sports data set

For UIUC-Sports data set [26], there are eight classes with totally 1579 images: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snow boarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images). For each class, the sizes of the instances in the same scene are very different, and the poses of the objects vary a lot. In addition, the background of each image is highly cluttered and discrepant. Even the number of the instances in the same category changes greatly. Some images from different classes have similar background (see in Fig. 5). We follow the common setup: 70 images per class are randomly selected as the training data, and 60 images per class for testing. Figs. 6 and 7 show the confusion matrices of our DLSM and DLMM method for image classification. Table 2[2] shows the performance of different methods. We notice that the classes of croquet and bocce have a high probability of being classified mistakenly, because these two classes are visually similar to each other. Furthermore, the classification rate by DLSM and DLMM is almost the same.

6.3. Scene 15 data set

For Scene 15 data set, there are 15 classes, with totally 4485 images. Each class varies from 200 to 400 images. The images contain not only indoor scenes, such as bedroom, living room, PARoffice, kitchen, and store, but also outdoor scenes, such as industrial, forest, mountain, tallbuilding, highway, street, opencountry, and so on (see in Fig. 8). We use an identical experimental setup as [1], where 100 images per class are randomly selected as the training data, and the rest for testing. Figs. 9 and 10 show the confusion matrices of our DLMM and DLSM method for image classification. Table 3 lists the comparisons of our two methods with previous work for image classification.

[1] For UIUC-Sports data set, we resize the maximum side to 400 due to the high resolution of original image.

[2] All the results of OCSVM and HIKVQ are based on step size 8 and without concatenated Sobel images.

Fig. 5. Example images from the UIUC-Sports data set. For each class, the instances in the same category are very different, the pose of the objects vary a lot, and the background of each is highly clutter and discrepancy. Some images from different classes have similar background.

	RockClimbing	badminton	bocce	croquet	polo	rowing	sailing	snowboarding
RockClimbing	93.54	0.00	3.33	2.92	1.25	0.83	0.00	3.75
badminton	0.00	95.21	1.67	1.04	1.46	0.42	0.00	1.04
bocce	0.63	1.67	71.46	12.92	2.71	2.50	0.63	5.00
croquet	0.21	0.83	14.79	77.29	1.88	0.83	3.33	1.04
polo	0.21	0.83	3.33	2.92	89.17	2.08	0.00	1.25
rowing	2.29	0.42	2.92	1.46	1.88	88.75	2.50	1.46
sailing	0.00	0.63	0.21	0.83	0.00	1.25	92.92	0.21
snowboarding	3.13	0.42	2.29	0.63	1.67	3.33	0.63	86.25

Fig. 6. Confusion matrix on UIUC-Sport data set (%) by DLSM.

	RockClimbing	badminton	bocce	croquet	polo	rowing	sailing	snowboarding
RockClimbing	93.13	0.00	3.54	2.92	1.25	0.83	0.00	3.33
badminton	0.00	95.42	2.08	0.83	1.04	0.42	0.00	1.25
bocce	0.63	1.46	71.46	13.96	2.50	2.50	0.63	5.42
croquet	0.21	0.83	14.37	76.88	1.88	0.63	3.33	1.04
polo	0.42	0.83	3.12	2.50	90.00	2.08	0.00	1.04
rowing	2.29	0.42	2.92	1.46	1.88	89.17	2.50	1.46
sailing	0.00	0.63	0.21	0.83	0.00	1.25	92.92	0.21
snowboarding	3.33	0.42	2.29	0.63	1.46	3.13	0.63	86.25

Fig. 7. Confusion matrix on UIUC-Sport data set (%) by DLMM.

Table 2
Performance comparison on UIUC-Sports data set.

Methods	Average classification rate (%)
HIKVQ [3]	81.87 ± 1.14
OCSVM [3]	81.33 ± 1.56
ScSPM [2]	82.74 ± 1.46
KScSPM [32]	84.92 ± 0.78
LScSPM [14]	85.31 ± 0.51
DLSM	86.82 ± 1.04
DLMM	$\mathbf{86.93 \pm 0.99}$

6.4. Caltech-101

Caltech-101 data set introduced in [29] contains 102 classes, one of which is the background. After removing the background class, the rest 101 classes with totally 8677 images are used for classification, with each class varying from 31 to 800 images. We follow the common experimental setup for this data set, where 15 and 30 images per category are selected as the training set, and the rest for the testing set (for training on 15 images per category, the maximum is 20 images per category for testing, and for training on 30 images per category, the maximum is 50 images

Fig. 8. Example images from the Scene 15 data set. The images contains not only indoor scene, but also outdoor scene.

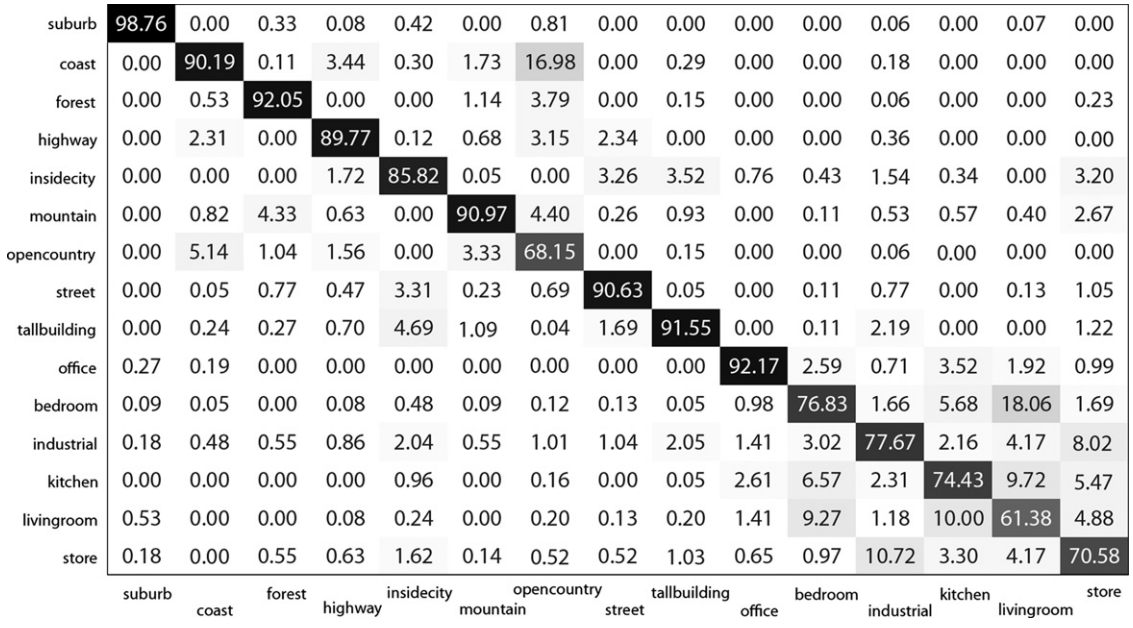

Fig. 9. Confusion matrix on Scene 15 data set (%) by DLSM.

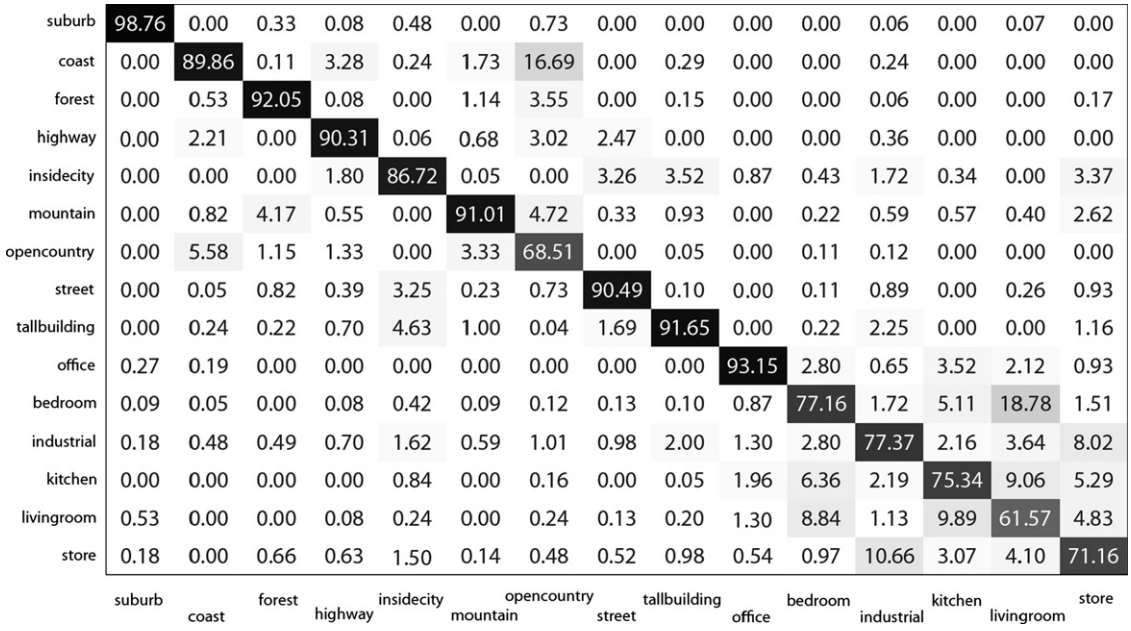

Fig. 10. Confusion matrix on Scene 15 data set (%) by DLMM.

Table 3
Performance comparison on scene 15 data set.

Methods	Average classification rate (%)
KSPM [1]	81.4 ± 0.5
KC [11]	76.7 ± 0.4
HIKVQ [3]	81.77 ± 0.49
OCSVM [3]	82.02 ± 0.54
ScSPM [2]	80.28 ± 0.93
KScSPM [32]	83.68 ± 0.61
LScSPM [14]	89.75 ± 0.50
DLSM	83.40 ± 0.44
DLMM	83.67 ± 0.49

Table 4
Performance comparison on Caltech-101 data set.

Methods	15 training	30 training
KSPM [1]	–	64.6 ± 0.8
KC [11]	–	64.1 ± 1.2
LLC[a] [12]	63.92 ± 0.46	70.63 ± 0.99
ScSPM [2]	67.0 ± 0.45	73.2 ± 0.54
DLSM	66.88 ± 0.53	74.39 ± 0.82
DLMM	$\mathbf{67.54 \pm 0.41}$	$\mathbf{74.87 \pm 0.67}$

[a] For LLC, we adopt the code of local feature coding provided by [12] and do experiment on our data set with single scale features and the size of dictionary 1024.

per category for testing). Table 4 shows the performance of different methods.

6.5. Caltech-256

Caltech-256 data set introduced in [30] contains 257 classes, one of which is the background. After removing the background class, the rest 256 classes with totally 29,780 images are used for classification. Due to much higher intra-class variability and higher object location variability compared with Caltech-101, Caltech-256 is a very challenging data set so far for image classification. We follow the common experimental setup for this data set: 15, 30, 45 and 60 training images per category and 15 testing images per category. Table 5 shows the performance of different methods.

Table 5
Performance comparison on Caltech-256 data set.

Methods	15 training	30 training	45 training	60 training
LLC[a] [12]	28.00 ± 0.36	33.34 ± 0.57	36.24 ± 0.37	38.08 ± 0.39
ScSPM [2]	27.73 ± 0.51	34.02 ± 0.35	37.46 ± 0.55	40.14 ± 0.91
KScSPM [32]	29.77 ± 0.14	35.67 ± 0.10	38.61 ± 0.19	40.30 ± 0.22
LScSPM [14]	30.00 ± 0.14	35.74 ± 0.10	38.54 ± 0.36	40.43 ± 0.38
DLSM	29.31 ± 0.58	35.12 ± 0.34	37.62 ± 0.57	39.96 ± 0.62
DLMM	$\mathbf{30.35 \pm 0.42}$	$\mathbf{36.22 \pm 0.33}$	$\mathbf{38.97 \pm 0.56}$	$\mathbf{41.09 \pm 0.44}$

[a] For LLC, we adopt the code of local feature coding provided by [12] and do experiment on our data set with single scale features and the size of dictionary 1024.

6.6. Experiments revisit

In this section, we revisit the experiments, and give the best parameter selection for balancing the weight among the fitting goodness, sparsity and the geometrical structure at first. Then, the comparison of DLSM and DLMM is also presented.

6.6.1. Parameter selection

The parameter α is the regularization parameter to control the tradeoff between the fitting goodness and the sparseness. With the increasing value of α, the codes become sparser and sparser, more and more salient, easy to distinguish; on the contrary, the reconstruction error becomes larger and larger, which will lead to inaccurate description of the codes. The parameter η is the regularization parameter to balance between the fitting goodness and the geometrical structure. With the increasing value of η, the geometrical structure becomes more and more reliable; the codes becomes more and more reasonable, and the reconstruction error becomes larger and larger. These two parameters are the competitors of reconstruction error.

First, we studied the effect of different η for DLMM and DLSM. Fig. 11(a) lists the performance when $\eta = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.6, 0.8, 1.0\}$ with $\alpha = 0.15$. As can be seen, when $\eta = 0.0$, DLSM

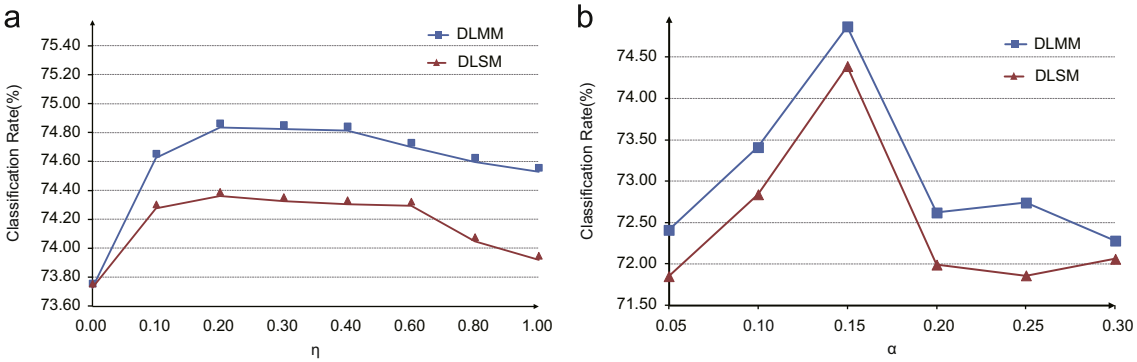

Fig. 11. Selection of parameters. (a) The classification rate under different η (Caltech-101) with $\alpha = 0.15$ and (b) the classification rate under different α (Caltech-101) with $\eta = 0.2$.

and DLMM are equivalent to sparse coding based dictionary learning method. With η growing, the classification rate increases. The best classification accuracy can be obtained when $\eta = 0.2$. After that, the performance starts to degenerate. When applying $\eta = 0.2$ to other data sets, we can also achieve the optimal results. Then, following the steps to obtain the optimal η, we fix $\alpha = \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$ with $\eta = 0.2$, and obtain the optimal $\alpha = 0.15$ (see Fig. 11(b)).

6.6.2. Comparison of DLSM and DLMM

According to the experimental results, we can see that, for the small size of data set, both DLSM and DLMM can achieve satisfying results, such as UIUC-sports data set and Scene 15 data set. However, with the increase of the size of the data and the number of the category, DLMM method becomes superior to DLSM. This is because, with the increase of the size of the data and the number of the category, the complexity of image patches is raising, so that single manifold cannot fit the cases well.

With the increasing number of the training data, the improvement of DLSM decreases faster than DLMM compared with ScSPM. When 60 training data per category is adopted, the classification accuracy is lower than ScSPM, which means that the single manifold cannot represent the real image features. Co-incidently, in [14], the author also concluded that as the number of training data increases, the improvement of LScSPM also decreases. When 60 training data per category is adopted, the classification rate merely exceeded 0.29%. However, our DLMM method exceeded 0.95%.

6.6.3. Comparisons on strong manifolds structures

To evaluate the effect of k-neighbors, we carry out an experiment on ORL face databases with strong manifolds structures. The ORL face data set contains 10 images for each of 40 human faces with each lying on the same manifold. The mean classification accuracy for each class is recorded (see in Table 6). From Table 6, the classification accuracy for the traditional sparse coding is 93%. The best classification accuracy for DLMM is 97%. And the best classification accuracy for DLSM is 95.5%. The classification accuracy for DLSM is high, because all the human faces are very similar, and single manifold may be approaching smooth. Obviously, the multiple manifolds structure is more reasonable than the single manifold structure.

From the comparison of k-neighbors, we can see that for DLMM, the classification accuracy is stable except when the number of neighbors is set to 5, because 5-neighborhood is too small to construct the multiple manifolds structures. For DLSM, the classification accuracy is unstable. When the number of neighbors is set to a large number, such as 150 or 199, the classification rate is less than the traditional sparse coding algorithm.

Table 6
Comparison on ORL face data set among different k-neighbors.

k-Neighbors	5	10	15	20	30	40	50
DLMM (%)	92.5	94	94	95.5	**97**	96	95.5
DLSM (%)	94	94	93.5	94	92.5	94.5	95
SC (%)	93						
k-neighbors	60	70	80	90	100	150	199
DLMM (%)	94.5	95.5	95	95	95	95	95.5
DLSM (%)	95.5	95	93.5	93.5	93.5	91.5	91.5

7. Conclusion

In this paper, we have proposed dictionary learning methods on manifolds for image classification. The methods consider the intrinsic geometrical structures identified by locally linear embedding on single manifold and sparse representation on multiple manifolds, respectively. Then the geometrical structures are embedded into sparse coding algorithm in an unified framework so as to preserve them during the sparse coding procedure. After that, a coordinate descent scheme is proposed to solve the optimization subproblems. The uncovered manifold structure makes it more congruent with image classification task. Experimental results on four benchmark data sets demonstrate that our proposed algorithm leads to more effective image representation and gains better classification performance. For the future, learning kernel based dictionary on manifolds will be carried out. And more rational topology will also be explored to improve the performance of image classification.

Acknowledgments

This work has been supported by the National Natural Science Foundation under Grants NNSF-61171118 and the Ministry of Education under Grants SRFDP-20110002110057.

We would like to thank Qianhaoze You for the suggestion on revision of this manuscript.

References

[1] A. Goldberg, X. Zhu, A. Singh, Z. Xu, R. Nowak, Multi-manifold semi-supervised learning, in: Proceedings of 12th International Conference on Artificial Intelligence and Statistics, 2009.

[2] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323.

[3] C. Chang, C. Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 27:1–27:27.

[4] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: Proceedings of the Nineth International Conference on Computer Vision, IEEE, vol. 2, 2003, pp. 1470–1477.
[5] J. Yang, S. Yang, Y. Fu, X. Li, T. Huang, Non-negative graph embedding, in: Proceedings of the 21rd International Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[6] J. Wu, J. Rehg, Beyond the euclidean distance: creating effective visual codebooks using the histogram intersection kernel, in: Proceedings of the 12th International Conference on Computer Vision, IEEE, 2009, pp. 630–637.
[7] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, D. Cai, Graph regularized sparse coding for image representation, IEEE Transactions on Image Processing 20 (5) (2011) 1327–1336.
[8] J. Lee, Introduction to Smooth Manifolds, vol. 218, Springer Verlag, 2003.
[9] J. Tenenbaum, V. De Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[10] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of the 22nd International Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1794–1801.
[11] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Proceedings of the 18th International Conference on Computer Vision and Pattern Recognition, IEEE, vol. 2, 2005, pp. 524–531.
[12] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Advances in Neural Information Processing Systems 14 (2001) 585–591.
[13] B. Shen, L. Si, Nonnegative matrix factorization clustering on multiple manifolds, in: Proceedings of the 24th AAAI Conference on Artificial Intelligence, 2010, pp. 575–580.
[14] J. van Gemert, C. Veenman, A. Smeulders, J. Geusebroek, Visual word ambiguity, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (7) (2010) 1271–1283.
[15] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proceedings of the 23rd International Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3360–3367.
[16] K. Ramamurthy, J. Thiagarajan, A. Spanias, Improved sparse coding using manifold projections, in: Proceedings of the 18th IEEE International Conference on Image Processing, IEEE, 2011, pp. 1237–1240.
[17] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, Advances in Neural Information Processing Systems 22 (2009) 2223–2231.
[18] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[19] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.
[20] Z. Lu, Y. Peng, Latent semantic learning by efficient sparse coding with hypergraph regularization, in: Proceedings of the 25th AAAI Conference on Artificial Intelligence, 2011.
[21] S. Gao, I. Tsang, L. Chia, Kernel sparse representation for image classification and face recognition, in: Proceedings of the 11th European Conference on Computer Vision, Springer, 2010, pp. 1–14.
[22] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, International Journal of Computer Vision 42 (3) (2001) 145–175.
[23] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories, in: Workshop of the 17th International Conference on Computer Vision and Pattern Recognition, vol. 12, 2004, p. 178.
[24] S. Gao, I. Tsang, L. Chia, P. Zhao, Local features are not lonely—Laplacian sparse coding for image classification, in: Proceedings of the 23rd International Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3555–3561.
[25] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: Proceedings of the 10th International Conference on Computer Vision, 2005, pp. 604–610.
[26] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of the 19th International Conference on Computer Vision and Pattern Recognition, IEEE, vol. 2, 2006, pp. 2169–2178.
[27] Y. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: Proceedings of the 23rd International Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2559–2566.
[28] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms, Advances in Neural Information Processing Systems 19 (2007) 801–808.
[29] L. Li, L. Fei-Fei, What, where and who? Classifying events by scene and object recognition, in: Proceedings of the 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–8.
[30] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, Belmont, MA, 1999.
[31] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset, Technical Report 7694, California Institute of Technology, 2007.
[32] B. Liu, Y. Wang, Y. Zhang, Y. Zheng, Discriminant sparse coding for image classification, in: Proceedings of the 37th International Conference on Acoustics, Speech and Signal Processing, IEEE, 2012, pp. 2193–2196.

Bao-Di Liu was born in Shandong, China. He received his bachelor degree and master degree in China University of Petroleum in 2004 and 2007 respectively. Currently, he is a PhD student from the Department of Electronic Engineering, Tsinghua University, China. His research interests include computer vision and machine learning.

Yu-Xiong Wang received the BS degree in Electronic Engineering from Beijing Institute of Technology (BIT), Beijing, China, in 2009. Currently, he is a master graduate student in the Department of Electronic Engineering at Tsinghua University, Beijing, China. His research interests include image processing, computer vision and machine learning.

Yu-Jin Zhang received the PhD degree in Applied Science from the State University of Liège, Liège, Belgium, in 1989. From 1989 to 1993, he was post-doc fellow and research fellow with the Department of Applied Physics and Department of Electrical Engineering at the Delft University of Technology, Delft, the Netherlands. In 1993, he joined the Department of Electronic Engineering at Tsinghua University, Beijing, China, where he is a professor of Image Engineering (since 1997). He is an active researcher in Image Engineering, with current interests on object segmentation from images and video, segmentation evaluation and comparison, moving object detection and tracking, face recognition, facial expression detection/classification, content-based image and video retrieval, information fusion for high-level image understanding, etc. He has authored more than 20 books and published more than 400 papers in the areas of image processing, image analysis, and image understanding. Professor Zhang is vice president of China Society of Image and Graphics and director of academic committee of the Society, and is a Fellow of SPIE.

Bin Shen is a PhD student in Computer Science, Purdue University. His research interests include machine learning and its applications to data mining, computer vision. Before that, he got his BS and MS in 2007 and 2009 respectively, both from EE, Tsinghua University, China.