Graph Regularized Sparse Coding for Image Representation

Miao Zheng, Jiajun Bu, *Member, IEEE*, Chun Chen, *Member, IEEE*, Can Wang, Lijun Zhang, Guang Qiu, and Deng Cai, *Member, IEEE*

***Abstract*—Sparse coding has received an increasing amount of interest in recent years. It is an unsupervised learning algorithm, which finds a basis set capturing high-level semantics in the data and learns sparse coordinates in terms of the basis set. Originally applied to modeling the human visual cortex, sparse coding has been shown useful for many applications. However, most of the existing approaches to sparse coding fail to consider the geometrical structure of the data space. In many real applications, the data is more likely to reside on a low-dimensional submanifold embedded in the high-dimensional ambient space. It has been shown that the geometrical information of the data is important for discrimination. In this paper, we propose a graph based algorithm, called graph regularized sparse coding, to learn the sparse representations that explicitly take into account the local manifold structure of the data. By using graph Laplacian as a smooth operator, the obtained sparse representations vary smoothly along the geodesics of the data manifold. The extensive experimental results on image classification and clustering have demonstrated the effectiveness of our proposed algorithm.**

***Index Terms*—Image classification, image clustering, manifold learning, sparse coding.**

I. Introduction

IN IMAGE processing, image representation plays a very important role. Researchers have long sought sparse and efficient representations of images. Sparse representations encode many of the images using only a few active coefficients, which make the encoding easy to interpret and reduce the computational cost. It has been proven useful in many applications [1]–[3]. To achieve sparse representations, many approaches have been developed, e.g., sparse PCA [4], sparse NMF [5]. One of the most typical methods is sparse coding, which has received a lot of attention in machine learning, signal processing and neuroscience [6]–[10].

Given an input data matrix, sparse coding aims to find a set of basis vectors (i.e., dictionary) that capture high-level semantics, and the sparse coordinates with respect to the dictionary. Sparse coding has several advantages for data representation. First, it yields sparse representations such that each data point is represented as a linear combination of a small number of basis vectors. Thus, these data points can be interpreted in a more elegant way. Second, sparse representations naturally make for an indexing scheme that would allow quick retrieval. Third, the sparse representation can be overcomplete, which offers a wide range of generating elements. Potentially, the wide range allows more flexibility in signal representation and more effectiveness at tasks like signal extraction and data compression. Finally, there is considerable evidence that biological vision adopts sparse representations in early visual areas [10], [11]. Benefit from these advantages, sparse coding has been studied in many applications, such as image restoration [12], [13] signal classification [14], face recognition [15] and image classification [16], [17].

Recently, various researchers [18]–[20] have considered the case when the data is drawn from sampling a probability distribution that has support on or near to a submanifold of the ambient space. Here, a d-dimensional submanifold of a Euclidean space $\mathbb{R}^M$ is a subset $\mathcal{M}^d \subset \mathbb{R}^M$ which locally looks like a flat d-dimensional Euclidean space [21]. In order to detect the underlying manifold structure, many manifold learning algorithms have been proposed, such as locally linear embedding (LLE) [19], ISOMAP [18], and Laplacian Eigenmap [20]. All these algorithms use the so-called locally invariant idea [22], i.e., the nearby points are likely to have similar embeddings. It has been shown that learning performance can be significantly enhanced if the geometrical structure is exploited and the local invariance is considered.

Motivated by recent progress in sparse coding and manifold learning, in this paper we propose a novel algorithm, called graph regularized sparse coding (GraphSC), which explicitly considers the local geometrical structure of the data. GraphSC builds a k-nearest neighbor graph to encode the geometrical information in the data. Using techniques from spectral graph theory, we use the graph Laplacian as a smooth operator to preserve the local manifold structure. Specifically, the graph Laplacian is incorporated into the sparse coding objective function as a regularizer. In this way, the obtained representations vary smoothly along the geodesics of the data manifold. By preserving locality, GraphSC can have more discriminating power

Manuscript received June 09, 2010; revised October 13, 2010; accepted October 13, 2010. Date of publication November 01, 2010; date of current version April 15, 2011. This work was supported by the China National Key Technology R&D Program under Grants 2008BAH26B00 and 2007BAH11B06, by the National Natural Science Foundation of China under Grants 60875044 and 90920303, and the National Basic Research Program of China (973 Program) under Grant 2011CB302206. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark (Hong-Yuan) Liao.

M. Zheng, J. Bu, C. Chen, C. Wang, and L. Zhang are with Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Zhejiang 310027, China (e-mail: cauthy@zju.edu.cn; bjj@zju.edu.cn; chenc@zju.edu.cn; wcan@zju.edu.cn; zljzju@zju.edu.cn).

G. Qiu is with Alibaba, Zhejiang 310052, China (e-mail: qiuguang@zju.edu.cn).

D. Cai is with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, 388 Yu Hang Tang Rd., Hangzhou, Zhejiang 310058, China (e-mail: dengcai@cad.zju.edu.cn).

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

Digital Object Identifier 10.1109/TIP.2010.2090535

1057-7149/$26.00 © 2010 IEEE

compared with traditional sparse coding algorithms and, hence, can facilitate machine learning tasks such as classification and clustering. The experimental results in this paper will show the effectiveness of GraphSC.

The rest of this paper is organized as follows: We review the related works on sparse coding in Section II. In Section III, we provide a brief description of sparse coding problem and the common methods to solve the sparse coding problem. Section IV introduces the GraphSC algorithm, as well as the optimization scheme, including learning sparse representations and learning the dictionary. The experimental results on image classification and clustering are presented in Section V. Finally, we conclude the paper in Section VI.

II. Related Works

Recently people have witnessed a growing interest in sparse coding. Several authors have suggested effective optimization, extensions and modifications of the original algorithm.

One of the weak points of sparse coding is the expensive computational cost for the optimization of the sparse coding problem. Several work has been performed to pursue methods for solving the optimization problem more effectively. An iterative soft-thresholding method was proposed in [15], which consists of the following simple iteration: take a Barzilai-Borwein step length in the negative gradient direction and then apply the soft-thresolding operator to the result. Lee *et al.* [7] proposed a feature-sign search method to reduce the nondifferentiable L1-norm problem to a L1-regularized least squares problem, which accelerates the optimization process.

Several authors try to design a more proper dictionary for sparse coding. Traditionally, the dictionary is chosen from standard bases (e.g., wavelet [23], curvelet [24], contourlets [25], and bandlets [2]), or even generated from random matrices [26], [27]. Recently, there are several novel works on the design of the dictionary. One of the most efficient methods is the K-SVD method [28]. The K-SVD is a way to learn a dictionary, instead of exploiting standard bases as described previously, that leads to sparse representations for the data. This algorithm uses either orthogonal matching pursuit (OMP) or basis pursuit (BP), as part of its iterative procedure for learning the dictionary.

Recently, several studies focus on developing the theoretical frameworks which combine sparse coding with classical machine learning methods. In [14], the authors tried to combine the linear discriminant analysis (LDA) with sparse representations, which incorporated reconstruction properties, discrimination power and sparsity for robust classification. In [16], the authors proposed a discriminative approach that effectively exploited the corresponding sparse signal decompositions in image classification tasks, and learned a shared dictionary and the discriminative models.

All of the previously mentioned studies focus on different aspects of the original sparse coding. However, they all fail to consider the geometrical structure in the data, which has been demonstrated the powerful discriminant ability in many applications [29], [30]. Several variants of sparse coding methods have been proposed to add some additional constraints for grouping sparse coding coefficients to capture the structure in the data. Locally-invariant sparse representations can be learned by adding additional spatial consistent constraint, which pools the sparse coefficients across overlapping windows [31]. Mairal *et al.* proposed simultaneous sparse coding by jointly decomposing groups of similar signals on subsets of the learned dictionary, which is implemented by adding a group-sparsity regularizer [32].

In this paper, we propose a novel sparse coding algorithm to exploit the geometrical information in the data by using the manifold assumption which has been shown effective in classification and clustering tasks [33]. It is important to note that the similar idea has also been proposed in [34]. The detailed optimization scheme is not clearly presented in [34] and the effectiveness of the proposed approach is only evaluated on the image classification task [34]. Our paper provides the detailed explanation of the optimization scheme and conducts a thorough experimental evaluation on the image clustering task. These are the complementary contributions made in our paper compared with [34].

III. Sparse Coding Problem

Given a data matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{R}^{n\times m}$, let $B = [\mathbf{b}_1, \ldots, \mathbf{b}_k] \in \mathbb{R}^{n\times k}$ be the dictionary matrix, where each $\mathbf{b}_i$ represents a basis vector in the dictionary, and $S = [\mathbf{s}_1, \ldots, \mathbf{s}_m] \in \mathbb{R}^{k\times m}$ be the coefficient matrix, where each column is a sparse representation for a data point. Each data point $\mathbf{x}_i$ can be represented as a sparse linear combination of those basis vectors in the dictionary. A good representation together with dictionary should minimize the empirical loss function, which can be represented as $\sum_{i=1}^{m} \|\mathbf{x}_i - B\mathbf{s}_i\|_p$. The typical norms used for measuring the loss function are the L_p norms for $p = 1, 2$ and ∞. Following [28] and [35], we concentrate on the case of $p = 2$.

Then, the objective function of sparse coding can be formulated as follows:

$$\min_{B,S} \|X - BS\|_F^2 + \beta \sum_{i=1}^{m} f(\mathbf{s}_i) \quad \text{s.t.} \|\mathbf{b}_i\|^2 \le c,\ i = 1, \ldots, k \tag{1}$$

where f is a function to measure the sparseness of $\mathbf{s}_i$ and $\|\cdot\|_F$ denotes the matrix Frobenius norm. A straightforward choice of f is the L_0 norm of the s, that is, $f(s_i) = \|s_i\|_0$, which counts the nonzero entries of s. However, while fixing the dictionary B, the minimization problem on coefficient S has been proven to be an NP-hard problem [36]. We, therefore, turn to approximations/relaxations of the problem instead. There are two common methods for approximately solving this problem, i.e., matching pursuit (MP) [37] and basis pursuit (BP) [38]. MP tries to find the solution one entry at a time in a greedy way, while BP makes a convex relaxation of the original problem by replacing the L_0 norm with an L_1 norm, which has been adopted more often recently [7], [16].

By following [7], [16], and [38], f is chosen as $f(s_i) = \|s_i\|_1$, rather than the L_0 norm. The objective function then becomes

$$\min_{B,S} \|X - BS\|_F^2 + \beta \sum_{i=1}^{m} \|s_i\|_1 \quad \text{s.t.} \|\mathbf{b}_i\|^2 \le c, \qquad i = 1, \ldots, k. \tag{2}$$

Although the objective function in (2) is convex in B only or S only, it is not convex in both variables together. A natural approach to solve this problem is to iteratively optimize the objective function (2) by minimizing over one variable while keeping the other one fixed. Thus, it becomes an ℓ_1-regularized least squares problem plus an ℓ_2-constrained least squares problem, which can both be solved efficiently by several optimization methods.

IV. Graph Regularized Sparse Coding (GraphSC)

In this section, we present our Graph regularized Sparse Coding (GraphSC) algorithm which takes into account the local manifold structure of the data space.

A. Objective Function

Recall that sparse coding tries to find a dictionary B and a sparse coefficient matrix S whose product can best approximate the original data matrix. The column vectors of B can be regarded as the basis vectors and each column of S is the new representation of each data point in this new space. One might further hope that the basis vectors can respect the intrinsic Riemannian structure, rather than ambient Euclidean structure. A natural assumption here could be that if two data points $\mathbf{x}_i, \mathbf{x}_j$ are close in the intrinsic geometry of the data distribution, then $\mathbf{s}_i$ and $\mathbf{s}_j$, the representations of this two points with respect to the new basis, are also close to each other. This assumption is usually referred to as manifold assumption [20], [39], which plays an essential role in developing various kinds of algorithms including dimensionality reduction algorithms [20], clustering algorithms [40], [41] and semisupervised learning algorithms [33], [42].

Given a set of n-dimensional data points $\mathbf{x}_1, \cdots, \mathbf{x}_m$, we can construct a nearest neighbor graph G with m vertices, where each vertex represents a data point. Let W be the weight matrix of G. If $\mathbf{x}_i$ is among the k-nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the k-nearest neighbors of $\mathbf{x}_i$, $W_{ij} = 1$, otherwise, $W_{ij} = 0$. We define the degree of $\mathbf{x}_i$ as $d_i = \sum_{j=1}^m W_{ij}$, and $D = \mathrm{diag}\,(d_1, \ldots, d_m)$.

Consider the problem of mapping the weighted graph G to the sparse representations S, a reasonable criterion for choosing a "good" map is to minimize the following objective function [20]:

$$\frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m(\mathbf{s}_i - \mathbf{s}_j)^2 W_{ij} = \mathrm{Tr}(\mathrm{SLS}^T) \tag{3}$$

where $L = D - W$ is the Laplacian matrix. By incorporating the Laplacian regularizer (3) into the original sparse coding, we can get the following objective function of GraphSC:

$$\begin{aligned}&\min_{B,S}\|X - BS\|_F^2 + \alpha\mathrm{Tr}(\mathrm{SLS}^T) + \beta\sum_{i=1}^m\|\mathbf{s}_i\|_1\\ &\text{s.t.}\|\mathbf{b}_i\|^2 \le c,\ i = 1, \ldots, k\end{aligned} \tag{4}$$

where $\alpha \ge 0$ is the regularization parameter.

Following the iteratively optimization method in [7], we divide the GraphSC algorithm into two steps: learning graph regularized sparse codes S while fixing the dictionary B, and learning dictionary B while fixing the coefficient matrix S.

B. Learning Graph Regularized Sparse Codes S

In this section, we discuss how to solve problem (4) by fixing the dictionary B. The problem (4) becomes

$$\min_S \|X - BS\|_F^2 + \alpha\mathrm{Tr}(\mathrm{SLS}^T) + \beta\sum_{i=1}^m\|\mathbf{s}_i\|_1. \tag{5}$$

Because problem (5) with ℓ_1-regularization is nondifferentiable when $\mathbf{s}_i$ contains values of 0, the standard unconstrained optimization methods can not be applied. Several approaches have been proposed to solve the problem of this form [43]–[47]. In the following, we introduce an optimization method based upon coordinate descent to solve this problem. It is easy to see that the problem (5) is convex, thus, the global minimum can be achieved.

We update each vector $\mathbf{s}_i$ individually, while holding all the other vectors constant. In order to solve the problem by optimizing over each $\mathbf{s}_i$, we should rewrite the problem (5) in a vector form.

The reconstruction error $\|X - BS\|_F^2$ can be rewritten as follows:

$$\sum_{i=1}^m\|\mathbf{x}_i - B\mathbf{s}_i\|^2. \tag{6}$$

The Laplacian regularizer $\mathrm{Tr}(\mathrm{SLS}^T)$ can be rewritten as follows:

$$\begin{aligned}\mathrm{Tr}(\mathrm{SLS}^T) &= \mathrm{Tr}\left(\sum_{i,j=1}^m L_{ij}\mathbf{s}_i\mathbf{s}_j^T\right)\\ &= \sum_{i,j=1}^m L_{ij}\mathbf{s}_j^T\mathbf{s}_i = \sum_{i,j=1}^m L_{ij}\mathbf{s}_i^T\mathbf{s}_j.\end{aligned} \tag{7}$$

Combining (6) and (7), the problem (5) can be rewritten as

$$\min\sum_{i=1}^m\|\mathbf{x}_i - B\mathbf{s}_i\|^2 + \alpha\sum_{i,j=1}^m L_{ij}\mathbf{s}_i^T\mathbf{s}_j + \beta\sum_{i=1}^m\|\mathbf{s}_i\|_1. \tag{8}$$

When updating $\mathbf{s}_i$, the other vectors $\{\mathbf{s}_j\}_{j\neq i}$ are fixed. Thus, we get the following optimization problem:

$$\begin{aligned}\min_{\mathbf{s}_i} f(\mathbf{s}_i) &= \|\mathbf{x}_i - B\mathbf{s}_i\|^2\\ &\quad + \alpha L_{ii}\mathbf{s}_i^T\mathbf{s}_i + \mathbf{s}_i^T\mathbf{h}_i + \beta\sum_{j=1}^k\left|s_i^{(j)}\right|\end{aligned} \tag{9}$$

where $\mathbf{h}_i = 2\alpha(\sum_{j\neq i} L_{ij}\mathbf{s}_j)$ and $s_i^{(j)}$ is the jth coefficient of $\mathbf{s}_i$.

Following the feature-sign search algorithm proposed in [7], the problem 9 can be solved as follows. In order to solve the nondifferentiable problem, we adopt a subgradient strategy, which uses subgradients of $f(\mathbf{s}_i)$ at nondifferentiable points. Firstly,

we define $h(\mathbf{s}_i) = ||\mathbf{x}_i - B\mathbf{s}_i||^2 + \alpha L_{ii}\mathbf{s}_i^T\mathbf{s}_i + \mathbf{s}_i^T\mathbf{h}_i$, then $f(\mathbf{s}_i) = h(\mathbf{s}_i) + \beta\sum_{j=1}^{k}|s_i^{(j)}|$. Recall that in nonsmooth optimizations, a necessary condition for a parameter vector to be a local minima is that the zero-vector is an element of the subdifferential, the set containing all subgradients at this parameter vector [48]. We define $\nabla_i^{(j)}|s_i|$ as the subdifferentiable value of the jth coefficient of s_i. If $|s_i^{(j)}| > 0$, then the absolute value function $|s_i^{(j)}|$ is differentiable, therefore, $\nabla_i^{(j)}|s_i|$ is given by the $\text{sign}(s_i^{(j)})$. If $s_i^{(j)} = 0$, then the subdifferentiable value $\nabla_i^{(j)}|s_i|$ is the set $[-1, 1]$. So, the optimality conditions for achieving the optimal value of $f(s_i)$ translate to

$$\begin{cases} \nabla_i^{(j)}h(s_i) + \beta\text{sign}\left(s_i^{(j)}\right) = 0, & \text{if } \left|s_i^{(j)}\right| > 0 \\ \left|\nabla_i^{(j)}h(s_i)\right| \le \beta, & \text{if } s_i^{(j)} = 0. \end{cases} \tag{10}$$

Then, we consider how to select the optimal subgradient $\nabla_i^{(j)}f(s_i)$ when the optimality conditions are violated, i.e., in the case that $|\nabla_i^{(j)}h(s_i)| > \beta$ if $s_i^{(j)} = 0$. When $s_i^{(j)} = 0$, we consider the first term in the previous expression $\nabla_i^{(j)}h(s_i)$. Suppose that $\nabla_i^{(j)}h(s_i) > \beta$. This means that $\nabla_i^{(j)}f(s_i) > 0$, regardless of the sign of $s_i^{(j)}$. In this case, in order to decrease $f(\mathbf{s}_i)$, we will want to decrease $s_i^{(j)}$. Since $s_i^{(j)}$ starts at zero, the very first infinitesimal adjustment to $s_i^{(j)}$ will take it negative. Therefore, for our purposes we can let $\text{sign}(s_i^{(j)}) = -1$. Similarly, if $\nabla_i^{(j)}h(s_i) < -\beta$, then we can effectively let $\text{sign}(s_i^{(j)}) = 1$.

To update $\mathbf{s}_i$, suppose we have known the signs of the $s_i^{(j)}$'s at the optimal value, then we can remove the ℓ_1-form on $s_i^{(j)}$ by replacing each term $|s_i^{(j)}|$ with either $s_i^{(j)}$ (if $s_i^{(j)} > 0$), $-s_i^{(j)}$ (if $s_i^{(j)} < 0$), or 0 (if $s_i^{(j)} = 0$). Thus, the problem (9) can be reduced to a standard, unconstrained quadratic optimization problem (QP), which can be solved by the linear system. The algorithmic procedure of learning graph regularized sparse codes is described in the following:

- for each $\mathbf{s}_i$, search for signs of $\{s_i^{(j)}\}_{j=1,\ldots,k}$,;
- solve the reduced QP problem to get the optimal $\mathbf{s}_i^*$ which minimizes the objective function (9);
- return the optimal coefficients matrix $S^* = [\mathbf{s}_1^*, \ldots, \mathbf{s}_m^*]$.

In the algorithm, we maintain an active set $\mathcal{A} \triangleq \{j | s_i^{(j)} = 0, |\nabla_i^{(j)}h(s_i)| > \beta\}$ for potentially nonzero coefficients and their corresponding signs $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_k]$ while updating each $\mathbf{s}_i$. Then, it systematically searches for the optimal active set and coefficient signs which minimize the objective function (9). In each activate step, the algorithm uses the zero-value whose violation of the optimality condition $|\nabla_i^{(j)}h(s_i)| > \beta$ is largest. The algorithm proceeds in a series of "feature-sign steps:" at each step, given a current value for the active set and the signs, it computes the analytical solution $\hat{\mathbf{s}}_i^{\text{new}}$ to the resulting unconstrained QP; then it updates the solution, the active set and the signs using an efficient discrete line search between the current solution and $\hat{\mathbf{s}}_i^{\text{new}}$. The detailed algorithmic procedure of learning graph regularized sparse codes, i.e., the matrix S, is stated in Algorithm 1.

Algorithm 1 Learning Graph regularized Sparse Codes

Input: A data set of m data points $X = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$, the dictionary B, the graph Laplacian matrix L, the parameters α and β.

1: **for all** i such that $1 \le i \le m$ **do**

2: **Initialize step**: $\mathbf{s}_i = \vec{0}, \boldsymbol{\theta} = \vec{0}$, and active set $\mathcal{A} = \emptyset$, where $\theta_j \in \{-1, 0, 1\}$ denotes $\text{sign}(s_i^{(j)})$.

3: **Activate step**: From zero coefficient of $\mathbf{s}_i$, select $j = \arg\max_j |\nabla_i^{(j)}h(\mathbf{s}_i)|$. Activate $s_i^{(j)}$ (add j to the active set) only if it locally improves the objective (9), namely:

$$\text{If} \nabla_i^{(j)}h(\mathbf{s}_i) > \beta, \text{ then set } \theta_j = -1, \mathcal{A} = \{j\} \cup \mathcal{A}.$$
$$\text{If} \nabla_i^{(j)}h(\mathbf{s}_i) < -\beta, \text{ then set} \theta_j = 1, \mathcal{A} = \{j\} \cup \mathcal{A}.$$

4: **Feature-sign step**:
(a). Let $\hat{B}$ be a submatrix of B that contains only the columns corresponding to the active set. Let $\hat{\mathbf{s}}_i$ and $\hat{\mathbf{h}}_i$ be subvectors of $\mathbf{s}_i$ and $\mathbf{h}$. Let $\hat{\boldsymbol{\theta}}$ be $\boldsymbol{\theta}$ corresponding to the active set.
(b). Compute the solution to the resulting unconstrained QP:

$$\min\ g(\hat{\mathbf{s}}_i) = ||\mathbf{x}_i - \hat{B}\hat{\mathbf{s}}_i||^2 + \alpha L_{ii}\hat{\mathbf{s}}_i^T\hat{\mathbf{s}}_i + \hat{\mathbf{s}}_i^T\hat{\mathbf{h}}_i + \beta\hat{\boldsymbol{\theta}}^T\hat{\mathbf{s}}_i \tag{11}$$

Let $(\partial g(\hat{\mathbf{s}}_i)/\partial\hat{\mathbf{s}}_i) = 0$, we can get the optimal value of $\mathbf{s}_i$ under the current active set:

$$\hat{\mathbf{s}}_i^{\text{new}} = (\hat{B}^T\hat{B} + \alpha L_{ii}I)^{-1}(\hat{B}^T\mathbf{x}_i - (\beta\hat{\boldsymbol{\theta}} + \hat{\mathbf{h}}_i)/2) \tag{12}$$

where I is the identity matrix.
(c). Perform a discrete line search on the closed line segment from $\hat{\mathbf{s}}_i$ to $\hat{\mathbf{s}}_i^{\text{new}}$: Check the objective value at $\hat{\mathbf{s}}_i^{\text{new}}$ and all points where any coefficient changes sign, and update $\hat{\mathbf{s}}_i$ (and the corresponding entries in $\mathbf{s}_i$) to the point with the lowest objective value.
(d). Remove zero coefficients of $\hat{\mathbf{s}}_i$ from the active set and update $\boldsymbol{\theta} = \text{sign}(\mathbf{s}_i)$.

5: **Check the optimality conditions step**:
Condition (a): Optimality condition for nonzero coefficients: $\nabla_i^{(j)}h(\mathbf{s}_i) + \beta\,\text{sign}\,(s_i^{(j)}) = 0, \forall s_i^{(j)} \neq 0$ If condition (a) is not satisfied, go to Step 4 (without any new activation); else check condition (b).
Condition (b): Optimality condition for zero coefficients: $|\nabla_i^{(j)}h(\mathbf{s}_i)| \le \beta, \forall s_i^{(j)} = 0$ If condition (b) is not satisfied, go to step 3; otherwise return $\mathbf{s}_i$ as the solution, redenoted as $\mathbf{s}_i^*$.

6: **end for**

Output: The optimal coefficient matrix $S^* = [\mathbf{s}_1^*, \ldots, \mathbf{s}_m^*]$.

C. Learning Dictionary B

In this section, we describe the method of learning the dictionary B, while fixing the coefficient matrix S. The problem becomes a least squares problem with quadratic constraints

$$\begin{aligned}&\min_B \|X - BS\|_F^2\\ &\text{s.t.} \|\mathbf{b}_i\|^2 \le c,\ i = 1, \ldots, k.\end{aligned} \tag{13}$$

There are many methods for solving this problem, such as gradient descent with iterative projection [49]. In this paper, we adopt the method that uses a Lagrange dual, which has been shown more efficient than gradient descent [7].

Let $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_k]$, and λ_i be the Lagrange multiplier associated with the ith inequality constraint $\|\mathbf{b}_i\|^2 - c \le 0$, then the Lagrange dual function of (13) is given by

$$\begin{aligned}g(\boldsymbol{\lambda}) &= \inf_B L(B, \boldsymbol{\lambda})\\ &= \inf_B \left(\|X - BS\|_F^2 + \sum_{i=1}^{k} \lambda_i (\|\mathbf{b}_i\|^2 - c) \right).\end{aligned} \tag{14}$$

Let Λ be the $k \times k$ diagonal matrix whose diagonal entry $\Lambda_{ii} = \lambda_i$ for all i. Then $L(B, \boldsymbol{\lambda})$ can be written as

$$\begin{aligned}L(B, \boldsymbol{\lambda}) &= \|X - BS\|_F^2 + \mathrm{Tr}(B^T B \Lambda) - c\mathrm{Tr}(\Lambda)\\ &= \mathrm{Tr}(X^T X) - 2\mathrm{Tr}(B^T X S^T)\\ &\quad + \mathrm{Tr}(S^T B^T B S) + \mathrm{Tr}(B^T B \Lambda) - c\mathrm{Tr}(\Lambda).\end{aligned} \tag{15}$$

The optimal solution B^* can be obtained by letting the first-order derivative of (15) equal to zero

$$B^* S S^T - X S^T + B^* \Lambda = 0. \tag{16}$$

Then, we have

$$B^* = X S^T (S S^T + \Lambda)^{-1}. \tag{17}$$

Substituting (17) into (15), the Lagrange dual function becomes

$$\begin{aligned}g(\boldsymbol{\lambda}) &= \mathrm{Tr}(X^T X) - 2\,\mathrm{Tr}(X S^T (S S^T + \Lambda)^{-1} S X^T)\\ &\quad - c\mathrm{Tr}(\Lambda) + \mathrm{Tr}((S S^T + \Lambda)^{-1} S X^T X S^T)\\ &= \mathrm{Tr}(X^T X) - \mathrm{Tr}(X S^T (S S^T + \Lambda)^{-1} S X^T)\\ &\quad - c\mathrm{Tr}(\Lambda).\end{aligned} \tag{18}$$

This leads to the following Lagrange dual function:

$$\begin{aligned}&\min_\Lambda \mathrm{Tr}(X S^T (S S^T + \Lambda)^{-1} S X^T) + c\mathrm{Tr}(\Lambda)\\ &\text{s.t.} \lambda_i \ge 0,\ i = 1, \ldots, k.\end{aligned} \tag{19}$$

This problem can be solved by using Newtons method or conjugate gradient. Let Λ^* be the optimal solution, then the optimal $B^* = X S^T (S S^T + \Lambda^*)^{-1}$. It is important to note that $S S^T + \Lambda$ is not guaranteed invertible. In practice, one can use pseudoinverse instead of directly computing the inverse.

V. Experimental Results

In this section, we present both image classification and clustering experiments on publicly available image data sets. For each experiment, we describe the information of data sets and the detailed settings. All the experimental results demonstrate the efficacy of our proposed algorithm.

TABLE I
Test Set Error Rates on USPS Handwritten Digits Dataset
(m Is the Size of Training Set)

m	100	500	1000	2000	5000	7291
OR	18.3	11.6	9.9	8.6	7.7	6.2
SC	20.4	12.0	9.6	8.2	7.1	5.5
GraphSC	**16.8**	**9.8**	**8.0**	**7.4**	**6.4**	**5.0**

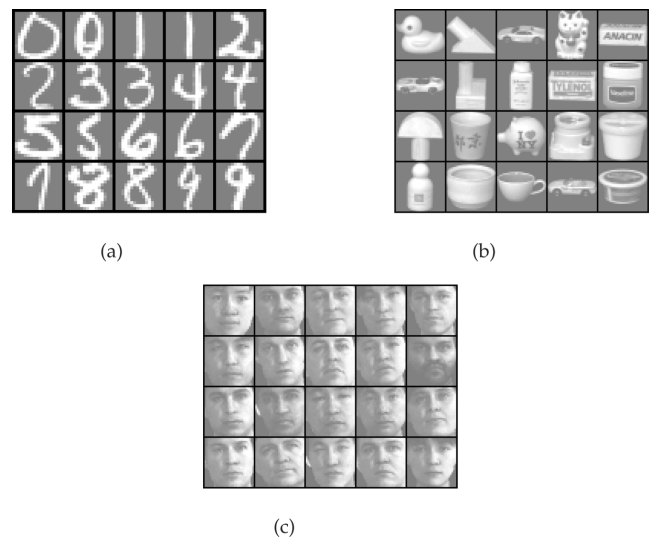


Fig. 1. Examples from USPS, COIL20 and CMU-PIE data sets. (a) USPS. (b) COIL20. (c) CMU-PIE.

A. Image Classification

For image classification, we present experiments on the benchmark USPS handwritten digits data set.[1] USPS is composed of 7291 training images and 2007 test images of size 16×16. Each image is represented by a 256-dimensional vector.

It is important to note that we need to compute the sparse representation for a new data point in the classification task. Let $X \in \mathbb{R}^{n \times m}$ denote the training data matrix and $W \in \mathbb{R}^{m \times m}$ denote the kNN graph matrix. We can use GraphSC to learn the dictionary matrix $B \in \mathbb{R}^{n \times k}$ and coefficient matrix $S \in \mathbb{R}^{k \times m}$. To compute the sparse representation of a new data point $\mathbf{x}_t$, we need to modify the graph matrix W. Without loss of generality, let $\widehat{X} = [X; \mathbf{x}_t] \in \mathbb{R}^{n \times (m+1)}$. The $\widehat{W} \in \mathbb{R}^{(m+1) \times (m+1)}$ can be constructed as

$$\widehat{W} = \begin{bmatrix} W & \mathbf{w} \\ \mathbf{w}^T & 0 \end{bmatrix} \tag{20}$$

where $\mathbf{w} \in \mathbb{R}^m$ is the k-nearest neighbors weight vector of $\mathbf{x}_t$ in X. Specifically, $w_i = 1$ if and only if $\mathbf{x}_i$ in the training set is among the k-nearest neighbors of $\mathbf{x}_t$. Now we can get the new Laplacian matrix which includes the new data point. By solving

[1]http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html

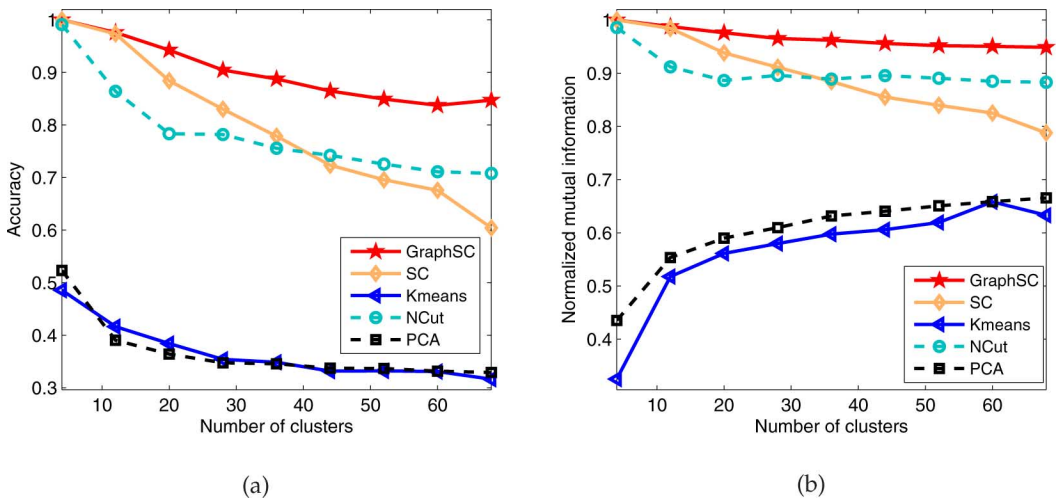


Fig. 2. (a) Accuracy. (b) Normalized mutual information versus. the number of clusters on CMU-PIE data set.

the optimization problem (9) (fixing B and S), we are able to compute the sparse representation $\mathbf{s}_t$ for the new data point $\mathbf{x}_t$.

We perform five-fold cross validation to find the best parameter pair for the original sparse coding (SC), which will be used afterward in GraphSC. The test values for the size of dictionary d are $\{32, 64, 128, 256, 512\}$, for the sparsity parameter β, $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Besides, there are another two parameters in GraphSC, the regularization parameter α and the number of nearest neighbors k. While fixing the best parameter pair (d, β), we perform five-fold cross validation to find the best pair for (α, k). The test values for the α are $\{0.01, 0.1, 1, 10, 100\}$, and for $k, \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We keep the best pair of parameters for GraphSC. For image classification task, we train linear SVM classifiers to evaluate the performance of different algorithms.

Table I shows the test results on the original representation (OR) and the sparse representation of the image data set obtained by SC and GraphSC. As can be seen, our algorithm GraphSC performs best in all cases. This certifies that GraphSC captures the most discriminative features of the images.

B. Image Clustering

For image clustering, we investigate the clustering performance of GraphSC on two real world image data sets, i.e., CMU-PIE face database and COIL20 image database.

We compare the following five algorithms for data clustering:

- K-means clustering algorithm (K-means);
- our proposed graph regularized sparse coding (GraphSC) + K-means;
- normalized cut (Ncut) [50] + K-means;
- principle component analysis (PCA) + K-means;
- sparse coding (SC) [7] + K-means.

Both SC and GraphSC algorithms learn sparse representations for the data points. Particularly, GraphSC explicitly takes into account the local manifold structure. Ncut is a spectrally-based algorithm which constructs an affinity matrix and uses its eigenvectors to represent the data. PCA is one of the most well known unsupervised dimensionality reduction algorithm which can be applied to remove the noise corresponding to the small eigenvalues of the data covariance matrix. In all the experiments, we first apply PCA to reducing the data dimensionality by keeping 98% information. The SC, GraphSC, and Ncut algorithms are then performed in the subspace.

1) Evaluation Metrics: We evaluate the clustering results by comparing the cluster label of each sample with its label provided by the data set. Two standard clustering metrics, the accuracy (AC) and the normalized mutual information metric $(\overline{\mathrm{MI}})$ are used to measure the clustering performance [51]. Given a data point $\mathbf{x}_i$, let r_i and s_i be the cluster label and the label provided by the data set, respectively. The AC is defined as follows:

$$\mathrm{AC} = \frac{\sum_{i=1}^{n} \delta(s_i, \mathrm{map}(r_i))}{n} \tag{21}$$

where n is the total number of samples and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and map (r_i) is the permutation mapping function that maps each cluster label r_i to the best label from the data set. The best mapping can be found by using the Kuhn-Munkres algorithm [52].

Let C denote the set of clusters obtained from the ground truth and C' obtained from our algorithm. Their mutual information metric $\mathrm{MI}(C, C')$ is defined as follows:

$$\mathrm{MI}(C, C') = \sum_{c_i \in C, c_j' \in C'} p(c_i, c_j') \cdot \log_2 \frac{p(c_i, c_j')}{p(c_i) \cdot p(c_j')} \tag{22}$$

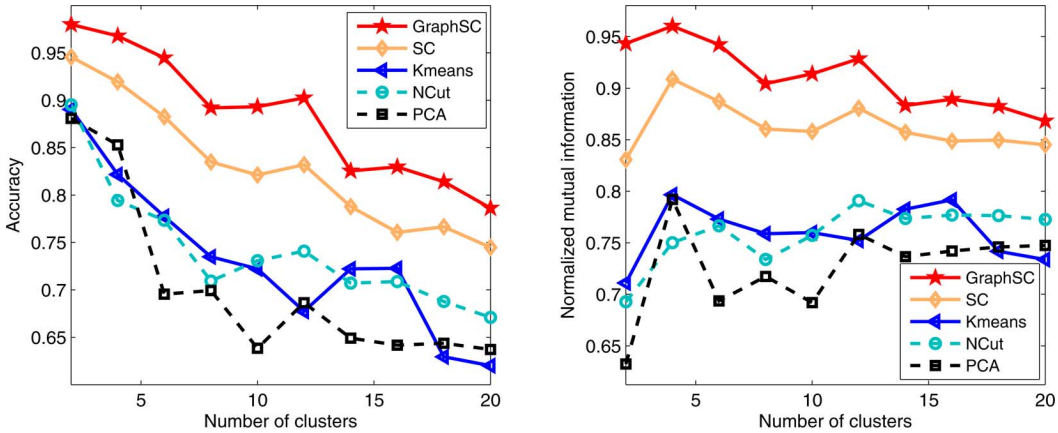
where $p(c_i)$ and $p(c_j')$ are the probabilities that a sample arbitrarily selected from the data set belongs to the clusters c_i and c_j', respectively, and $p(c_i, c_j')$ is the joint probability that the arbitrarily selected sample belongs to the clusters c_i as well as c_j' at the same time. In our experiments, we use the normalized mutual information $\overline{\mathrm{MI}}$ as follows:

$$\overline{\mathrm{MI}}(C, C') = \frac{\mathrm{MI}(C, C')}{\max(H(C), H(C'))} \tag{23}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $\overline{\mathrm{MI}}(C, C')$ ranges from 0 to 1.

TABLE II
CLUSTERING RESULTS ON CMU-PIE DATA SET (c IS THE NUMBER OF CLUSTERS)

c	Accuracy(%)					Normalized Mutual Information(%)				
	K-means	PCA	NCut	SC	GraphSC	K-means	PCA	NCut	SC	GraphSC
4	48.6	52.3	99.1	100	**100**	32.5	43.5	98.6	100	**100**
12	41.6	39.0	86.4	97.3	**97.6**	51.8	55.4	91.2	98.4	**98.8**
20	38.4	36.4	78.3	88.4	**94.3**	56.1	59.0	88.6	93.8	**97.6**
28	35.4	34.8	78.2	83.0	**90.5**	58.0	61.0	89.6	91.1	**96.5**
36	34.9	34.6	75.6	77.9	**88.8**	59.7	63.2	88.9	88.4	**96.2**
44	33.2	33.7	74.2	72.4	**86.5**	60.6	64.1	89.5	85.5	**95.6**
52	33.2	33.7	72.6	69.6	**84.9**	62.0	65.1	89.1	84.0	**95.2**
60	33.1	33.2	71.1	67.6	**83.8**	65.8	65.9	88.5	82.5	**95.1**
68	31.7	32.9	70.8	60.4	**84.7**	63.3	66.6	88.3	78.8	**94.9**
Avg.	36.7	36.7	78.5	79.6	**90.1**	56.6	60.4	90.3	89.2	**96.7**



(a) (b)

Fig. 3. (a) Accuracy. (b) Normalized mutual information versus the number of clusters on COIL20 data set.

$\overline{\mathrm{MI}} = 1$ if the two sets of clusters are identical, and $\overline{\mathrm{MI}} = 0$ if the two sets are independent.

2) Clustering on CMU-PIE Face Database: The CMU-PIE[2] face database contains 68 subjects with 41 368 face images as a whole. The size of each image is 32×32, with 256 grey levels per pixel. Thus, each image is represented by a 1024-dimensional vector. In this experiment, we fix the pose and expression. Thus, for each subject, we have 21 images under different lighting conditions.

We carry out the clustering experiments with the cluster number (c) ranging from 4 to 68. For each c except 68, 20 test runs are conducted on different randomly chosen clusters, and the final performance scores are obtained by averaging over the 20 tests. For each test, we first apply each one of the compared algorithms to learn a new representation for the data, and then apply K-means in the new representation space. The K-means is repeated 50 times with different initializations and the best result in terms of the objective function of K-means is recorded.

After PCA projection, the dimensionality is reduced to 64. Since several recent works on sparse coding have advocated the use of overcomplete representations for images, in which the dimensionality of the feature vectors is larger than the dimensionality of the input [28], [53], we use 128 basis vectors in the SC and GraphSC algorithms. For Ncut, we use c eigenvectors where c is the number of clusters, as suggested by previous work [40].

Fig. 2 shows the plots of clustering accuracy and normalized mutual information versus the number of clusters. As can be seen, our GraphSC algorithm consistently outperforms the other algorithms, especially when the number of clusters is large. The detailed clustering results are summarized in Table II. The average clustering accuracy and normalized mutual information for GraphSC is 90.1% and 96.7%, respectively. Compared with SC, our algorithm achieves 10.5% and 7.5% improvement in accuracy and normalized mutual information, respectively.

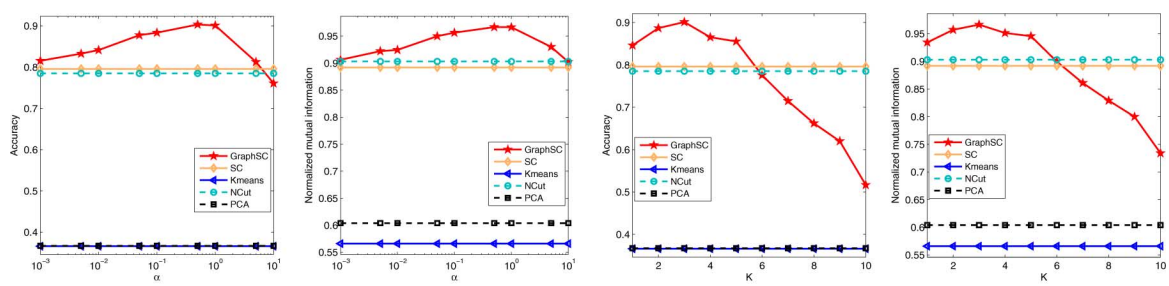
3) Clustering on COIL20 Image Database: The COIL20 image library[3] contains 1 440 32×32 gray scale images of

[2]http://www.ri.cmu.edu/projects/project_418.html

[3]http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php

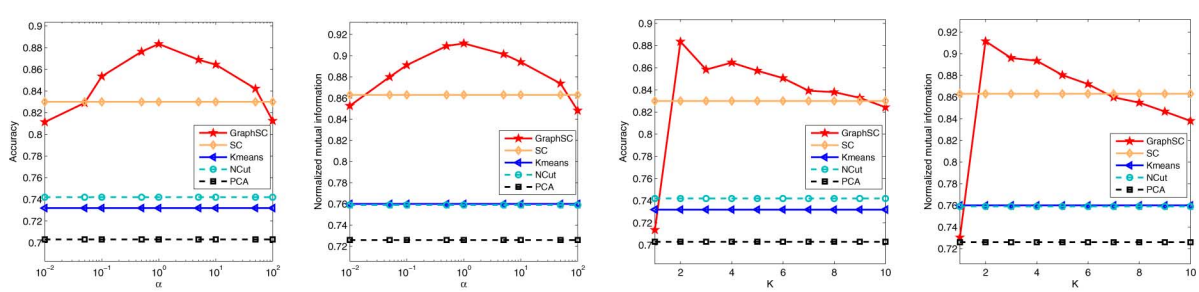
TABLE III
CLUSTERING RESULTS ON COIL20 IMAGE LIBRARY (c IS THE NUMBER OF CLUSTERS)

c	Accuracy(%)					Normalized Mutual Information(%)				
	K-means	PCA	NCut	SC	GraphSC	K-means	PCA	NCut	SC	GraphSC
2	89.0	88.1	89.5	94.5	**98.0**	71.1	63.2	69.3	83.1	**94.3**
4	82.2	85.3	79.4	91.9	**96.8**	79.7	79.2	75.0	90.9	**96.0**
6	77.7	69.5	77.3	88.3	**94.5**	77.3	69.4	76.6	88.7	**94.2**
8	73.5	69.9	70.9	83.5	**89.2**	75.9	71.7	73.4	86.0	**90.4**
10	72.2	63.9	73.1	82.1	**89.3**	76.0	69.2	75.7	85.8	**91.4**
12	67.8	68.7	74.1	83.2	**90.2**	75.2	75.8	79.1	88.0	**92.9**
14	72.2	64.9	70.7	78.8	**82.5**	78.3	73.6	77.4	85.7	**88.3**
16	72.3	64.2	70.9	76.1	**83.0**	79.2	74.2	77.7	84.9	**88.9**
18	62.9	64.4	68.8	76.7	**81.4**	74.2	74.6	77.6	85.0	**88.2**
20	62.0	63.7	67.1	74.5	**78.6**	73.4	74.7	77.3	84.5	**86.8**
Avg.	73.2	70.3	74.2	83.0	**88.4**	76.0	72.6	75.9	86.3	**91.1**



(a) (b) (c) (d)

Fig. 4. Clustering performance with different values of the regularization parameter (α) and the number of nearest neighbors (k) on CMU-PIE face database. (a) Accuracy versus α. (b) NMI versus α. (c) Accuracy versus k. (d) NMI versus k.



(a) (b) (c) (d)

Fig. 5. Clustering performance with different values of the regularization parameter (α) and the number of nearest neighbors (k) on COIL20 image database. (a) Accuracy versus α. (b) NMI versus α. (c) Accuracy versus k. (d) NMI versus k.

20 objects (72 images per object). The images of each object were taken 5° apart as the object is rotated on a turntable. Each image is represented by a 1024-dimensional vector.

The experiment setup is essentially the same as before. We carry out the experiments with the cluster number ranging from 2 to 20. For this data set, the dimensionality after PCA projection is 175. Therefore, we use 256 basis vectors.

Fig. 3 and Table III show the clustering performance of the five algorithms. As can be seen, our GraphSC algorithm performs the best in all the cases. SC performs the second best. These results indicate that by encoding geometrical information in the sparse representations, the learning performance can be significantly enhanced.

4) Model Selection on Clustering: In our GraphSC algorithm, there are two parameters, that are, the regularization parameter α and the number of nearest neighbors (k) for graph construction. We empirically set α to 1 and k to 3 in the previous experiments. In this section, we study the influence of different choices of α and k.

We vary α while fixing k, and vary k while fixing α. From Figs. 4 and 5, we observe that the performance of GraphSC is significantly better than other algorithms over a large range of both α and k.

VI. Conclusion

The authors present a novel sparse coding method called GraphSC that explicitly considers the manifold structure of the data space. By introducing a graph Laplacian regularizer into the traditional sparse coding objective function, we can obtain a set of basis vectors, and new sparse representations which capture the intrinsic geometrical information in the data. The experimental results on image classification and clustering have demonstrated that our proposed algorithm can have better discriminating power and significantly enhance the classification and clustering performance.

References

[1] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 1475–1490, Nov. 2004.

[2] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423–438, Apr. 2005.

[3] J. Starck, M. Elad, and D. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1570–1582, Oct. 2005.

[4] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," *SIAM Review*, vol. 49, no. 3, p. 434, 2007.

[5] P. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, 2004.

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

[7] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," *Adv. Neural Inf. Process. Syst.*, vol. 20, pp. 801–808, 2007.

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

[9] M. Lewicki and T. Sejnowski, "Coding time-varying signals using sparse," *Shift-Invariant Represent.*, pp. 730–736, 1999.

[10] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[11] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vis. Rese.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[12] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, 2006.

[13] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[14] K. Huang and S. Aviyente, "Sparse representation for signal classification," *Adv. Neural Inf. Process. Syst.*, vol. 19, pp. 609–616, 2007.

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

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," *Adv. Neural Inf. Process. Syst.*, vol. 21, pp. 1033–1040, 2009.

[17] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 759–766.

[18] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

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

[20] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Adv. Neural Inf. Process. Syst.*, vol. 15, pp. 585–592, 2002.

[21] J. M. Lee, *Introduction to Smooth Manifolds*. New York: Springer-Verlag, 2002.

[22] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1735–1742.

[23] S. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic, 1999.

[24] E. Candès and D. Donoho, "Recovering edges in ill-posed inverse problems: Optimality of curvelet frames," *Ann. Statist.*, vol. 30, no. 3, pp. 784–842, 2002.

[25] M. Do and M. Vetterli, "Framing pyramids," *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2329–2342, Sep. 2003.

[26] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[27] E. Candès, "Compressive sampling," in *Proc. Int. Congr. Math.*, 2006, pp. 1433–1452.

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

[29] D. Cai, X. Wang, and X. He, "Probabilistic dyadic data analysis with local and global consistency," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 105–112.

[30] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Trans. Knowl. Data Eng.*, 2011, to be published.

[31] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. Le-Cun, "Learning invariant features through topographic filter maps," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1605–1612.

[32] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2272–2279.

[33] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[34] S. Gao, I. Tsang, L. Chia, and P. Zhao, "Local features are not lonely-Laplacian sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3555–3561.

[35] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.

[36] B. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.

[37] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[38] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1999.

[39] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 63–72.

[40] A. Y. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, 2001, vol. 14, pp. 849–856.

[41] J. Liu, D. Cai, and X. He, "Gaussian mixture model with local consistency," in *Proc. 24th Conf. Artif. Intell.*, 2010, pp. 512–517.

[42] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Adv. Neural Inf. Process. Syst.*, vol. 16, pp. 595–602, 2003.

[43] V. Roth, "The generalized LASSO," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 16–28, Jan. 2004.

[44] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[45] G. Andrew and J. Gao, "Scalable training of L 1-regularized log-linear models," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 33–40.
[46] S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast, incremental feature selection by gradient descent in function space," *J. Mach. Learn. Res.*, vol. 3, pp. 1333–1356, 2003.
[47] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for l1 regularization: A comparative study and two new approaches," in *Proc. 18th Eur. Conf. Mach. Learn.*, 2007, pp. 286–297.
[48] R. Fletcher, *Practical Methods of Optimization*. Hoboken, NJ: Wiley, 1987.
[49] Y. Censor and S. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*. New York: Oxford Univ. Press, 1997.
[50] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[51] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
[52] L. Lovász and M. Plummer, *Matching Theory*. Amsterdam, The Netherlands: North Holland, 1986.
[53] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Adv. Neural Inf. Process. Syst.*, 2006, vol. 19, pp. 1137–1144.

Miao Zheng received the B.S. degree in computer science from Zhejiang University, Zhejiang, China, in 2008, and is currently pursuing the Ph.D. degree in computer science at the same university.

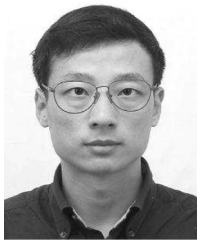
His research interests include machine learning, information retrieval, and data mining.

Jiajun Bu (M'06) received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Zhejiang, China, in 1995 and 2000, respectively.

He is currently a Professor in the College of Computer Science, Zhejiang University. His research interests include embedded system, data mining, information retrieval, and mobile database.

Chun Chen (M'06) received the B.S. degree in mathematics from Xiamen University, Xiamen, China, in 1981, and the M.S. and Ph.D. degrees in computer science from Zhejiang University, Zhejiang, China, in 1984 and 1990, respectively.

He is currently a Professor in the College of Computer Science, Zhejiang University. His research interests include information retrieval, data mining, computer vision, computer graphics, and embedded technology.



Can Wang received the B.S. degree in economics, the M.S. degree in computer science, and the Ph.D. degree from Zhejiang University, Zhejiang, China, in 1995, 2003, and 2009, respectively.

He is currently a Faculty Member in the College of Computer Science, Zhejiang University. His research interests include information retrieval, data mining, and machine learning.

Lijun Zhang received the B.S. degree in computer science from Zhejiang University, Zhejiang, China, in 2007, and is currently pursuing the Ph.D. degree in computer science at the same university.

His research interests include machine learning, information retrieval, and data mining.

Guang Qiu received the Ph.D. degree in computer science from Zhejiang University, Zhejiang, China, in 2010.

He is currently a researcher in Alibaba. His research interests include data mining, machine learning, and information retrieval.

Deng Cai (S'07–M'08) received he received the B.S. and M.S. degrees in automation from Tsinghua University, Beijing, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer science from University of Illinois, Urbana Champaign, in 2009.

He is currently an Associate Professor in the State Key Laboratory, CAD&CG, College of Computer Science, Zhejiang University, Zhejiang, China. His research interests include machine learning, data mining, and information retrieval.