# Large Scale Spectral Clustering Via Landmark-Based Sparse Representation

Deng Cai, *Member, IEEE*, and Xinlei Chen, *Student Member, IEEE*

***Abstract*—Spectral clustering is one of the most popular clustering approaches. However, it is not a trivial task to apply spectral clustering to large-scale problems due to its computational complexity of $O(n^3)$, where $n$ is the number of samples. Recently, many approaches have been proposed to accelerate the spectral clustering. Unfortunately, these methods usually sacrifice quite a lot information of the original data, thus result in a degradation of performance. In this paper, we propose a novel approach, called landmark-based spectral clustering, for large-scale clustering problems. Specifically, we select $p$ ($\ll n$) representative data points as the landmarks and represent the original data points as sparse linear combinations of these landmarks. The spectral embedding of the data can then be efficiently computed with the landmark-based representation. The proposed algorithm scales linearly with the problem size. Extensive experiments show the effectiveness and efficiency of our approach comparing to the state-of-the-art methods.**

***Index Terms*—Acceleration, bipartite, graph, landmarks, scalability, singular value decomposition, sparse coding, spectral clustering.**

## I. Introduction

CLUSTERING is one of the fundamental problems in data mining, pattern recognition, and many other fields. A series of methods have been proposed over the past decades [1], [20], [24], [28]. Among them, spectral clustering, a class of methods which are based on eigen-decomposition of matrices, often yields more superior experimental performance compared to other algorithms [35]. While many clustering algorithms are based on Euclidean geometry and consequently place limitations on the shape of the clusters, spectral clustering can adapt to a wider range of geometries and detect nonconvex patterns and linearly nonseparable clusters [13], [31].

Despite its good performance, spectral clustering is limited in its applicability to large-scale problems due to its high computational complexity. The general spectral clustering method needs to construct an adjacency matrix and calculate the eigen-decomposition of the corresponding Laplacian matrix [6]. Both of these two steps are computational expensive. For a data set consisting of $n$ data points, the above two steps will have time complexities of $O(n^2)$ and $O(n^3)$, which is an unbearable burden for large-scale applications in information retrieval, data mining, etc.

In recent years, much effort has been devoted for accelerating the spectral clustering algorithm. A natural option is finding the methods to reduce the computational cost of the eigen-decomposition of the graph Laplacian. Fowlkes *et al*. [14] adopted the classical Nyström method for efficiently computing an approximate solution of the eigenproblem. Furthermore, Li *et al*. [23] proposed a scalable Nyström scheme by using the randomized low-rank matrix approximation algorithms, making the algorithm even faster. Another option is to perform a reduction in the data size beforehand. Shinnou and Sasaki [36] replaced the original data set with a relatively small number of data points, and the follow-up operations are performed on the adjacency matrix corresponding to the smaller set. Based on a similar idea, Yan *et al*. [38] provided a general framework for fast approximate spectral clustering. Sakai and Imiya [34] used another variant which is based on random projection and sampling. Chen *et al.* [4] and Liu *et al.* [26] introduced a sequential reduction algorithm based on the observation that some data points converge to their true embedding quickly, so that an early stop strategy will speed up decomposition. However, their idea can only tackle binary clustering problems and have to resort to a hierarchical scheme for multiway clustering.

In the last decades, sparse coding [9], [32] method is proposed as a matrix factorization technique to model the human visual cortex, and later has been shown useful for applications in many fields such as visual neuroscience [22] and image restoration [12], [30]. Since sparse representations encode many data points using only a few active coefficients, the coding result is easy to interpret and the computational cost is reduced. Many sparse versions of classical algorithms have been developed, e.g., sparse principle component analysis [41], sparse nonnegative matrix factorization [19]. Moreover, a large amount of research has been devoted to seeking efficient optimization algorithms for sparse coding [7], [16], [21].

Inspired by the recent progress on sparse coding [9], [32] and scalable semi-supervised learning [27], we propose a scalable spectral clustering method termed landmark-based spectral clustering (LSC) in this paper. Specifically, LSC generates $p$ ($\ll n$) representative data points as the landmarks and represent the remaining data points as linear combinations of

Manuscript received July 19, 2012; revised June 2, 2014, July 21, 2014, and August 1, 2014; accepted September 1, 2014. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2013CB336500 and in part by the National Nature Science Foundation of China under Grant 61222207 and Grant 91120302. This paper was recommended by Associate Editor J. Huang.

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

X. Chen is with Language Technology Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: enderchen@cs.cmu.edu).

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/TCYB.2014.2358564

**Algorithm 1** Spectral Clustering

**Input:**

$n$ data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^m$;

Cluster number $k$;

**Output:**

$k$ clusters;

1: Construct a sparse affinity matrix $W \in \mathbb{R}^{n\times n}$ between data points, with the degree matrix $D \in \mathbb{R}^{n\times n}$ calculated according to (1);

2: Compute the smallest $k$ eigenvectors of $L = D^{-1/2}(D - W)D^{-1/2}$ (or largest $k$ eigenvectors of $D^{-1/2}WD^{-1/2}$), denoted by $Q = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k]$;

3: Each row of $Q$ is a data point and apply $k$-means to get the clusters.

these landmarks. The spectral embedding of the data can then be efficiently computed with the landmark-based representation. The overall proposed algorithm scales linearly with the problem size. Extensive experiments show the effectiveness and efficiency of our approach compared to the state-of-the-art methods.

The rest of this paper is organized as follows. In Section II, we provide a background knowledge of spectral clustering and several popular methods which are designed for speeding up the spectral clustering, along with a review for the sparse coding techniques. Our LSC method is introduced in Section III. After it, in Section IV, we present a theoretical analysis and a mathematical validation of the proposed algorithm. The experimental results on a number of real world data sets are reported in Section V. Finally, we provide the concluding remarks and suggestions for future work in Section VI.

## II. Background and Related Work

Given a set of data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$, spectral clustering first constructs an undirected graph $\mathcal{G} = (V, E)$ represented by its adjacency matrix $W = (w_{ij})_{i,j=1}^n$, where $w_{ij} \geq 0$ denotes the similarity (affinity) between $\mathbf{x}_i$ and $\mathbf{x}_j$. The degree matrix $D$ is a diagonal matrix whose entries are row (or column, since $W$ is symmetric) sums of $W$

$$D_{ii} = \sum_{j=1}^{n} w_{ij}. \tag{1}$$

Let $L = D - W$, which is called graph Laplacian [6]. Spectral clustering then uses the top $k$ eigenvectors of $L$ (or, the normalized Laplacian $D^{-1/2}LD^{-1/2}$) corresponding to the $k$ smallest eigenvalues[1] as the low dimensional (with dimensionality $k$) representations of the original data. Finally, the traditional $k$-means method [18] is applied to obtain the clusters.

The process of standard spectral clustering is listed in Algorithm 1. Due to the high complexity of the graph construction ($O(n^2)$) and the eigen-decomposition ($O(n^3)$), it is not easy to apply spectral clustering on large-scale data sets and henceforth a number of methods have been developed in the literature to accelerate the standard algorithm.

### A. Efficient Spectral Clustering Methods

A natural way to handle the scalability issue of spectral clustering is using the sampling technique. The basic idea is using preprocessing to reduce the data size.

Yan *et al*. [38] proposed the k-means-based approximate spectral clustering (KASP) method. It firstly performs $k$-means on the data set with a large cluster number $p$. Then, the traditional spectral clustering is applied on the $p$ cluster centers. The data point is assigned to the cluster as its nearest center.

Shinnou and Sasaki [36] adopted a slightly different way to reduce the data size. Their approach firstly applies $k$-means on the data set with a large cluster number $p$. It then removes those data points which are close to the centers (with predefined distance threshold). The centers are called committees in their algorithm. The traditional spectral clustering is applied on the remaining data points plus the cluster centers. Those removed data points are assigned to the cluster as their nearest centers. In the experiments, we named this approach committees-based spectral clustering (CSC).

Another way to handle the scalability issue of spectral clustering is reducing the computational cost of the eigen-decomposition step. Fowlkes *et al*. [14] applied the Nyström method to accelerate the eigen-decomposition. Given an $n \times n$ matrix, Nyström method compute the eigenvectors of a $p \times p$ ($p \ll n$) sub-matrix (randomly sampled from the original matrix). The calculated eigenvectors are used to estimate an approximation of the eigenvectors of the original matrix. Zhang *et al*. [40] provide an error analysis of this method and use $k$-means to sample the sub-matrix. Recent development [23] of this method performs an approximate singular value decomposition (SVD) on the inner sub-matrix of the column set sampled from the input graph by using the randomized low-rank matrix approximation algorithms, which speed up the decomposition even more.

All these approaches used the sampling idea. Some key data points are selected to represent the whole data set. Although this idea is very effective, quite a lot of information of the detailed structure of the data is lost in the sampling step.

### B. Matrix Factorization and Sparse Coding

The matrix factorization technique has become very popular in recent years for data representation. It tries to "compress" the data by finding a set of basis vectors and the representation with respect to the basis for each data point. Let $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m\times n}$ be the data matrix, matrix factorization can be mathematically defined as finding two matrices $U \in \mathbb{R}^{m\times p}$ and $Z \in \mathbb{R}^{p\times n}$ whose product can best approximate $X$ [25]

$$X \approx UZ.$$

Each column of $U$ can be regarded as a basis vector which captures the higher-level features in the data and each column of $Z$ is the $p$-dimensional representation of the original inputs with respect to the new basis. A common way to measure the

[1]It is easy to check that the eigenvectors of $D^{-1/2}LD^{-1/2}$ corresponding to the smallest eigenvalues are the same as the eigenvectors of $D^{-1/2}WD^{-1/2}$ corresponding to the largest eigenvalues [31].

approximation is by Frobenius norm of a matrix $\|\cdot\|$. Thus, the matrix factorization can be defined as the optimization problem as follows:

$$\min_{U,Z} \|X - UZ\|^2. \tag{2}$$

The solution $Z$ matrix of the above optimization problem is usually dense. Since each basis vector (column vector of $U$) can be regarded as a concept, a dense matrix $Z$ indicates that each data point is a combination of all the concepts. This is contrary to our common knowledge since most of the data points only include several semantic concepts.

Sparse coding [9], [32] is a recently popular matrix factorization method trying to solve this issue. Sparse coding adds the sparse constraint on $Z$, more specifically, on each column of $Z$, in the optimization problem (2)

$$\min_{U,Z} \|X - UZ\|^2 + \alpha f(Z) \tag{3}$$

where $f$ is a function which can measure the sparsity of each column of $Z$ (e.g., $l_1$ norm) and $\alpha$ is a coefficient controlling the sparsity penalty. In this way, SC can learn a sparse representation. SC 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, the 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 over-complete, 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 [33].

## III. LSC

In this section, we introduce our LSC for large scale spectral clustering. The basic idea of our approach is designing an efficient way for graph construction and Laplacian matrix eigen-decomposition. Specifically, we try to design the affinity matrix which has the property as follows:

$$W = \hat{Z}^T \hat{Z}$$

where $\hat{Z} \in \mathbb{R}^{p\times n}$ is sparse and $p \ll n$. If we can efficiently derive $\hat{Z}$, we can thus compute eigenvectors of the graph Laplacian in $O(p^3 + p^2 n)$ without explicitly storing the whole affinity matrix in the main memory, which saves a lot of computation load and storage space.

Our approach is motivated from the recent progress on sparse coding [9], [32] and scalable semi-supervised learning [27].

### A. Landmark-Based Sparse Coding

Each column of the solution matrix $Z$ in the optimization problem (3) is a $p$-dimensional representation of the original input with respect to the new basis. Thus, $Z^T Z$ is a similarity matrix of the data with respect to the new representation and naturally can be used as the affinity matrix to speed up spectral clustering. However, solving the optimization problem (3) is very time consuming, particularly when $X$ is dense. In fact, most of the existing approaches compute $U$ and $Z$ iteratively. Apparently, these approaches cannot be used for speeding up spectral clustering in large-scale applications.

To address this issue, we notice that the basis vectors (column vectors of $U$) have the same dimensionality with the original data points. We can treat the basis vectors as the landmark points of the data set. The most efficient way to select landmark points from a data set is random sampling. Besides random selection, we can also use the $k$-means algorithm to first cluster all the data points and then use the cluster centers as the landmark points. For efficiency consideration, we will focus on the random selection method although the comparison between random selection and $k$-means-based landmark selection is presented in our empirical study.

Suppose we already have the landmark matrix $U$, we can solve the optimization problem (3) to compute the representation matrix $Z$. By fixing $U$, the optimization problem becomes a constraint (sparsity constraints) linear regression problem. There are many algorithms [11] which can solve this problem. However, these optimization approaches are still time consuming in real applications. In our approach, we simply use Nadaraya–Watson kernel regression [17] to compute the representation matrix $Z$ [27]. A detailed analysis will be covered later.

For any data point $\mathbf{x}_i$, we find its approximation $\hat{\mathbf{x}}_i$ by

$$\hat{\mathbf{x}}_i = \sum_{j=1}^{p} z_{ji} \mathbf{u}_j \tag{4}$$

where $\mathbf{u}_j$ is $j$th column vector of $U$ and $z_{ji}$ is $ji$th element of $Z$. A natural assumption here is that $z_{ji}$ should be larger if $\mathbf{x}_i$ is closer to $\mathbf{u}_j$. We can emphasize this assumption by setting the $z_{ji}$ to zero as $\mathbf{u}_j$ is not among the $r$ ($\leq p$) nearest neighbors of $\mathbf{x}_i$. This restriction naturally leads to a sparse representation matrix $Z$. Let $U_{\langle i\rangle} \in \mathbb{R}^{m\times r}$ denote a sub-matrix of $U$ composed of $r$ nearest landmarks of $\mathbf{x}_i$. We compute $z_{ji}$ as

$$z_{ji} = \frac{K(\mathbf{x}_i, \mathbf{u}_j)}{\sum_{j'\in\langle i\rangle} K(\mathbf{x}_i, \mathbf{u}_{j'})} \quad j \in \langle i\rangle \tag{5}$$

where $K(\cdot)$ is a kernel function. The Gaussian or heat kernel

$$K(\mathbf{x}_i, \mathbf{u}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{u}_j\|^2}{2h^2}} \tag{6}$$

is one of the most commonly used in the literature, where $h$ is the bandwidth.

### B. Spectral Analysis on Landmark-Based Graph

We have the landmark-based sparse representation $Z \in \mathbb{R}^{p\times n}$ now and we simply compute the graph affinity matrix as

$$W = \hat{Z}^T \hat{Z} \tag{7}$$

where $\hat{Z} = \hat{D}^{-1/2} Z$, $\hat{D}$ is an $p \times p$ diagonal matrix whose entries are row sums of $Z$ ($\hat{D}_{ii} = \sum_j Z_{ij}$). Note that in the

previous section, each column of $Z$ sums up to 1 and thus according to (1)

$$D_{ii} = \sum_{j=1}^{n} w_{ij} = \sum_{j=1}^{n}\sum_{l=1}^{p} \frac{z_{li}z_{lj}}{\hat{D}_{ll}} = \sum_{l=1}^{p} z_{li}\frac{\sum_{j=1}^{n} z_{lj}}{\hat{D}_{ll}} = \sum_{l=1}^{p} z_{li} = 1.$$

Thus, the affinity matrix $W$ in (7) is automatically normalized.

Let the SVD [15] of $\hat{Z}$ be as follows:

$$\hat{Z} = A\Sigma B^T \tag{8}$$

where $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_p)$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$ are the singular values of $\hat{Z}$, $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p] \in \mathbb{R}^{p\times p}$ and $\mathbf{a}_i$'s are called left singular vectors, $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_p] \in \mathbb{R}^{n\times p}$ and $\mathbf{b}_i$'s are called right singular vectors. It is easy to check that the column vectors of $B$ are the eigenvectors of matrix $W = \hat{Z}^T\hat{Z}$; the column vectors of $A$ are the eigenvectors of matrix $\hat{Z}\hat{Z}^T$; and $\sigma_i^2$ are the eigenvalues.

Since the size of matrix $\hat{Z}\hat{Z}^T$ is $p \times p$, we can take advantage of it by first computing $A$ within $O(p^3)$ time. $B$ can then be computed as

$$B^T = \Sigma^{-1}A^T\hat{Z}. \tag{9}$$

The overall time is $O(p^3 + p^2n)$, which is a significant reduction from $O(n^3)$ considering $p \ll n$.

### C. Computational Complexity Analysis

In this section, we provide a computational complexity analysis of our algorithm. Suppose we have $n$ data points and we use $p$ landmarks as basis vectors. Let us take a look at each step.

1) If we adopt random sampling to select the landmarks, the time complexity can be ignored comparing to the follow-up steps. On the other hand, if we use $k$-means to select the landmarks, we need additional $O(tpn)$ time, where $t$ is the number of iterations in $k$-means.
2) To construct the affinity matrix $W$, we only need to compute the sparse representation matrix $Z$ according to (5). For each data point, we need to find the $r$ nearest neighbors from the $p$ landmarks. Thus, we need $O(pn)$ time in this step.
3) The step to generate the left singular vectors of $\hat{Z}$ takes a time boundary of $O(p^3)$, and we need another $O(p^2n)$ to derive right singular vectors according to (9). The overall time in finding the eigenvectors of $W$ is thus $O(p^3 + p^2n)$.

We summarize our algorithm in Algorithm 2 and the computational complexity in Table I. For the sake of comparison, Table I also lists several other popular accelerating spectral clustering methods. We use LSC-R to denote our method with random landmark selection and LSC-K to denote our method with $k$-means landmark selection.

## IV. Theoretical Analysis

In this section, we present a theoretical analysis of the proposed algorithm. First, we will discuss the relationship between LSC and bipartite graph normalized cut [8], [39]. Second, an error estimation is conducted to show that our method can preserve quite a lot information of the original data set, so it is reasonable to outperform other state-of-the-art methods. Finally, we provide an asymptotic property of LSC which connects our method to the traditional spectral clustering.

**Algorithm 2** LSC

**Input:**

$n$ data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^m$;
Cluster number $k$;

**Output:**

$k$ clusters;

1: Produce $p$ landmark points using $k$-means or random selection;
2: Construct a sparse affinity matrix $Z \in \mathbb{R}^{p\times n}$ between data points and landmark points, with the affinity calculated according to (5);
3: Compute the first $k$ eigenvectors of $\hat{Z}\hat{Z}^T$ where $\hat{Z} = \hat{D}^{-1/2}Z$, denoted it by $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k]$;
4: Compute $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k]$ according to (9);
5: Each row of $B$ is a data point and apply $k$-means to get the clusters.

TABLE I
Time Complexity of Accelerating Methods

| Method | Pre-process | Construction | Decomposition |
|---|---|---|---|
| KASP | $O(tpn)$ | $O(p^2)$ | $O(p^3)$ |
| CSC | $O(tpn)$ | $O(p^2)$ | $O(p^3)$ |
| Nyström | / | $O(pn)$ | $O(p^3 + pn)$ |
| LSC-R | / | $O(pn)$ | $O(p^3 + p^2n)$ |
| LSC-K | $O(tpn)$ | $O(pn)$ | $O(p^3 + p^2n)$ |

* $n$: # of points; $p$: # of landmarks / centers / sampled points; $t$: # of iterations in $k$-means.
** The final clustering is $O(tkn)$ for each algorithm, with $k$ denote the number of clusters.

### A. Bipartite Graph Interpretation

The matrix $Z$ unveils a tight affinity measure between the data points and the landmarks. To explicitly capture the data-landmark relationship, Liu *et al*. [27] introduced a relevant term in graph theory, bipartite [6]. Formally, a bipartite $\mathcal{B} = (V, U, E)$ consists of three components: $V$ includes the nodes $\mathbf{x}_i$ representing the data points; $U$ includes the nodes $\mathbf{u}_j$ representing the landmarks; and $E$ contains the edges between $V$ and $U$ [6]. Given a bipartite $\mathcal{B}$, bipartite graph normalized cut [8], [39] seeks to simultaneously partition the data points and the landmarks based on the spectral graph theory.

The affinity matrix of the whole bipartite graph $W_B$ can then be written as [8]

$$W_B = \begin{pmatrix} \mathbf{0} & Z^T \\ Z & \mathbf{0} \end{pmatrix}. \tag{10}$$

To partition the bipartite, the optimization task can be formalized as a generalized eigenvalue problem with suitable relaxation [35]

$$L_B\mathbf{q} = (D_B - W_B)\mathbf{q} = \lambda D_B\mathbf{q} \tag{11}$$

where $D_B$ is the degree matrix of $W_B$.

Put (10) in (11), we get

$$\begin{pmatrix} \mathbf{0} & Z^T \\ Z & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{pmatrix} = (1-\lambda) \begin{pmatrix} D_1 & \mathbf{0} \\ \mathbf{0} & D_2 \end{pmatrix} \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{pmatrix}$$

in which $D_1$ is an $n \times n$ diagonal matrix whose entries are column sums of $Z$ ($D_{1,jj} = \sum_i Z_{ij}$) and $D_2$ is an $p \times p$ diagonal matrix whose entries are row sums of $Z$ ($D_{2,ii} = \sum_j Z_{ij}$). Break the block matrix form into parts, the equation above can be rewritten as the following equations:

$$Z^T \mathbf{q}_2 = (1-\lambda) D_1 \mathbf{q}_1$$
$$Z \mathbf{q}_1 = (1-\lambda) D_2 \mathbf{q}_2.$$

Let $\mathbf{b} = D_1^{1/2} \mathbf{q}_1$ and $\mathbf{a} = D_2^{1/2} \mathbf{q}_2$, and after variable substitution, we have

$$D_1^{-1/2} Z^T D_2^{-1/2} \mathbf{a} = (1-\lambda) \mathbf{b}$$
$$D_2^{-1/2} Z D_1^{-1/2} \mathbf{b} = (1-\lambda) \mathbf{a}.$$

These are the exact equations that define the SVD of the normalized matrix $\hat{Z} = D_2^{-1/2} Z D_1^{-1/2}$. Particularly, $\mathbf{a}$ and $\mathbf{b}$ are the left and right singular vectors and $1-\lambda$ is the corresponding singular value [39].

As described in Section III, in LSC, we have $D_1 = I$ and $D_2 = \hat{D}$, so $\hat{Z} = D_2^{-1/2} Z D_1^{-1/2} = \hat{D}^{-1/2} Z$. Therefore, what we have done in the graph decomposition step is exactly the bipartite graph normalized cut [8], [39]. It strongly verifies the validity of our graph construction and decomposition process.

### B. Error Estimation for Sparse Representation

In LSC, we learn a sparse representation $\hat{\mathbf{x}}$ with some landmarks for each input $\mathbf{x}$ as in (4). Here we will provide an error estimation of this sparse representation.

$\mathbf{x}_i$ is represented by several landmarks, Proposition 1 puts an upper-bond to the estimation error.

*Proposition 1:* Assume all the $\mathbf{u}_j$ involved to estimate $\mathbf{x}_i$ is within a range of $\epsilon_i > 0$ of $\mathbf{x}_i$ using $l_2$ distance, then we have

$$\|\mathrm{BIAS}(\hat{\mathbf{x}}_i)\| = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\| < \epsilon_i \tag{12}$$

where $\hat{\mathbf{x}}_i$ is given by (4).

*Proof:* Based on (5), $z_{ji}$ sums up to 1. We have

$$\begin{aligned} \mathrm{BIAS}\left(\hat{\mathbf{x}}_i\right) &= \hat{\mathbf{x}}_i - \mathbf{x}_i \\ &= \sum_{j \in \langle i \rangle} z_{ji} \mathbf{u}_j - \mathbf{x}_i \\ &= \sum_{j \in \langle i \rangle} z_{ji} \left(\mathbf{u}_j - \mathbf{x}_i\right). \end{aligned}$$

Combined with the fact that $z_{ji}$ is nonnegative

$$\begin{aligned} \|\mathrm{BIAS}\left(\hat{\mathbf{x}}_i\right)\| &\le \sum_{j \in \langle i \rangle} z_{ji} \|\mathbf{u}_j - \mathbf{x}_i\| \\ &\le \epsilon_i \sum_{j \in \langle i \rangle} z_{ji} = \epsilon_i. \end{aligned}$$

■

So the overall upper bond of the estimation bias is

$$\mathrm{BIAS}(U) = \sum_{i=1}^{n} \epsilon_i \tag{13}$$

which is defined as a function of selected landmarks given the data set.

Finding out the optimal solution of $U$ with regard to (13) is hard and time consuming, so we turn to other heuristic methods to generate landmarks like $k$-means and random sampling. In a sense, the results obtained by these methods can be perceived as perturbation to the optimal landmarks.

Let the optimal landmarks be the mean vectors to random variables $\hat{U} = [\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \ldots, \hat{\mathbf{u}}_p]$ generated by an oracle method. Assume landmarks $\hat{\mathbf{u}}_j$ generated by other methods are independently distributed, we can treat the perturbation as adding a noise component $\xi_j$ with zero mean [37]

$$\mathbf{u}_j = \hat{\mathbf{u}}_j + \xi_j \tag{14}$$

for each $j = 1, 2, \ldots, p$. To make the problem tractable, we further assume the following.

1) All $\xi_j$'s are independent with a distribution with mean zero ($\mathbf{E}(\xi_j) = 0$) and satisfy $\mathbf{E}(\xi_j^T \xi_j) < \rho_j$.
2) The variance of $\xi_j$ is small relative to that of the original data set, so that the nearest neighbor set of $\mathbf{x}_i$, i.e., $U_{\langle i \rangle}$ is preserved.
3) $\epsilon_i$ is much smaller than kernel bandwidth $h$, i.e., $\epsilon_i \ll h$.

Now we derive the upper-bond for the perturbed landmarks, see Proposition 2.

*Proposition 2:* Under the assumptions discussed above, given kernel function described in (6) and kernel bandwidth $h \gg \epsilon_i$, the mean square error (MSE) of perturbed landmarks $U$ on estimated data point $\hat{\mathbf{x}}_i$ satisfies

$$\mathrm{MSE}(\hat{\mathbf{x}}_i) \le \frac{\rho}{r} + \epsilon_i^2 \left(1 + \frac{\rho}{2rh^2}\right) + O\left(\frac{\epsilon_i^4}{rh^4}\right) \tag{15}$$

where $\rho$ is the maximum value over $\rho_j$ and merely related to the distribution of $\xi_j$, $j = 1, 2, \ldots, p$.

*Proof:* The proof is threefold. First, the MSE of the estimated $\mathbf{x}_j$ can be divided into two components [2]

$$\begin{aligned} \mathrm{MSE}\left(\hat{\mathbf{x}}_i\right) &= \mathbf{E}\left[\hat{\mathbf{x}}_i - \mathbf{x}_i\right]^2 = \mathbf{E}\left[\left(\hat{\mathbf{x}}_i - \mathbf{E}\left[\hat{\mathbf{x}}_i\right]\right) + \left(\mathbf{E}\left[\hat{\mathbf{x}}_i\right] - \mathbf{x}_i\right)\right]^2 \\ &= \left\|\mathbf{x}_i - \mathbf{E}\left[\hat{\mathbf{x}}_i\right]\right\|^2 + \left\{\mathbf{E}\left[\hat{\mathbf{x}}_i\right] - \hat{\mathbf{x}}_i\right\}^T \left\{\mathbf{E}\left[\hat{\mathbf{x}}_i\right] - \hat{\mathbf{x}}_i\right\} \\ &= \left\|\mathbf{x}_i - \mathbf{E}\left[\hat{\mathbf{x}}_i\right]\right\|^2 + \mathrm{VAR}\left(\hat{\mathbf{x}}_i\right) \\ &= \left\|\sum_{j \in \langle i \rangle} z_{ji} \left(\mathbf{E}\left[\mathbf{u}_j\right] - \mathbf{x}_i\right)\right\|^2 + \mathrm{VAR}\left(\hat{\mathbf{x}}_i\right) \\ &= \left\|\sum_{j \in \langle i \rangle} z_{ji} \left(\hat{\mathbf{u}}_j - \mathbf{x}_i\right)\right\|^2 + \mathrm{VAR}\left(\hat{\mathbf{x}}_i\right) \\ &= \mathrm{BIAS}^2\left(\hat{\mathbf{x}}_i\right) + \mathrm{VAR}\left(\hat{\mathbf{x}}_i\right) \end{aligned}$$

where $\mathrm{BIAS}(\hat{\mathbf{x}}_i)$ has an upper-bond that is already verified in Proposition 1, so we merely prove the limit on $\mathrm{VAR}(\hat{\mathbf{x}}_i) = \left\{\mathbf{E}[\hat{\mathbf{x}}_i] - \hat{\mathbf{x}}_i\right\}^T \left\{\mathbf{E}[\hat{\mathbf{x}}_i] - \hat{\mathbf{x}}_i\right\}$. Note here it does not stand for a covariance matrix, but rather a scalar to measure the error caused by perturbation to the optimal landmarks.

Second, we give an upper-bond to measure $z_{ji}$ for the follow-up estimation for $\mathrm{VAR}(\hat{\mathbf{x}}_i)$

$$\max_j z_{ji} \le \frac{\epsilon_i^2}{2rh^2} + O\left(\frac{\epsilon_i^4}{h^4}\right). \tag{16}$$

Given $\epsilon_i \ll h$, we have

$$z_{ji} = \frac{K_h(\mathbf{x}_i, \mathbf{u}_j)}{\sum_{j' \in \langle i \rangle} K_h(\mathbf{x}_i, \mathbf{u}_{j'})} = \frac{e^{-\frac{|\mathbf{u}_j - \mathbf{x}_i|^2}{2h^2}}}{\sum_{j'=1}^{r} e^{-\frac{|\mathbf{u}_{j'} - \mathbf{x}_i|^2}{2h^2}}} \leq \frac{1}{r} e^{\frac{\epsilon_i^2}{2h^2}} \approx \frac{1}{r} + \frac{\epsilon_i^2}{2rh^2} + O\left(\frac{\epsilon_i^4}{h^4}\right).$$

Third, we prove that $\mathrm{VAR}(\hat{\mathbf{x}}_i)$ has the property

$$\begin{aligned}\mathrm{VAR}\left(\hat{\mathbf{x}}_i\right) &= \left\{\mathbf{E}\left[\hat{\mathbf{x}}_i\right] - \hat{\mathbf{x}}_i\right\}^T \left\{\mathbf{E}\left[\hat{\mathbf{x}}_i\right] - \hat{\mathbf{x}}_i\right\} \\ &= \mathbf{E}\left[\left(\sum_{j \in \langle i \rangle} z_{ji}\xi_j\right)^T \left(\sum_{j \in \langle i \rangle} z_{ji}\xi_j\right)\right] \\ &= \sum_{j \in \langle i \rangle}\left[z_{ji}^2 \mathbf{E}\left(\xi_j^T \xi_j\right)\right] = \sum_{j \in \langle i \rangle}\left[z_{ji}^2 \rho_j\right] \\ &\leq \rho \sum_{j \in \langle i \rangle} z_{ji}^2 \leq \rho \max_j z_{ji} \\ &\leq \frac{\rho}{r} + \frac{\rho \epsilon_i^2}{2rh^2} + O\left(\frac{\epsilon_i^4}{h^4}\right).\end{aligned}$$

Finally, combining $\|\mathrm{BIAS}(\hat{\mathbf{x}}_i)\|$ and $\mathrm{VAR}(\hat{\mathbf{x}}_i)$ together, we obtain the final result. ■

From the above analysis, we can make several preliminary observations.

1) For a single point $\mathbf{x}_i$, with perturbation on landmarks, as long as the perturbation to the optimal landmarks (where $\epsilon_i$ is small) are not so devastating that the neighborhood of it is still preserved, the MSE will probably be small.
2) For a single point $\mathbf{x}_i$, the accuracy of estimation will improve as more landmarks are selected to represent the actual value of $\mathbf{x}_i$. However, since $\epsilon_i$ may get larger as more nearby landmarks are added, we should maintain a balance in practice.
3) Generally, the distribution of the overall landmarks can have a strong impact on the estimation outcome since a terrible pattern may result in a huge $\epsilon_i$ for some data point, which is quadratic to the upper-bond. Since $k$-means-based landmark selection aims to minimize an objective function
$$\sum_{\mathbf{c}_j \in \mathcal{C}} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \mathbf{c}_j\|^2$$
where $C_j$ denotes cluster the point $\mathbf{x}_i$ is assigned, $\mathbf{c}_j$ is the corresponding centroid for the cluster (later selected as a landmark), and $\mathcal{C}$ is the set of all centroids. Therefore, it tends to outperform the random sampling.
4) A better result would be generated if we select landmarks by minimizing the objective function
$$\sum_{\mathbf{x}_i \in \mathcal{X}} \sum_{j \in \langle i \rangle} \|\mathbf{x}_i - \mathbf{u}_j\|^2$$
where $\langle i \rangle$ denotes an index set for the nearest $r$ landmarks of $\mathbf{x}_i$. For a fixed $p$ and $r$, how to solve the optimization problem efficiently and effectively still needs further research work.

### C. Asymptotic Property and Other Weighting Options for Z

In LSC, the original data matrix $X \in \mathbb{R}^{m \times n}$ is represented as a low dimensional sparse matrix $Z \in \mathbb{R}^{p \times n}$ with respect to the $p$ landmarks. The affinity graph matrix $W = \hat{Z}^T \hat{Z}$ does not explicitly belong to any of the traditional ones like $r$-NN graph. Nevertheless, we have the following proposition to discover the relationship between our landmark-based graph and the typical graphs used in spectral clustering, which in a sense, demonstrates the asymptotic property of LSC as $p$ gets closer to $n$.

*Proposition 3:* As $p$ converges to $n$, the clustering result of LSC-K and LSC-R will converge to that of the standard spectral clustering.

*Proof:* First, it is easy to check that as $p$ converges to $n$, the landmarks $\mathbf{u}_j$ (in either LSC-K or LSC-R) will converge to the data points $\mathbf{x}_i$, which means

$$\lim_{p \to n} Z = W_s, \lim_{p \to n} \hat{Z} = \hat{W}_s$$

where $W_s(\hat{W}_s)$ denotes the (normalized) affinity matrix constructed in the standard spectral clustering.

Second, let the SVD of $\hat{Z}$ be as follows:

$$\hat{Z} = A \Sigma B^T$$

and let the Eigenvalue decomposition of $\hat{W}_s$ be

$$\hat{W}_s = Q \Lambda Q^T.$$

Since $\hat{Z}$ converges to $\hat{W}_s$, which is a symmetric matrix with a boundary ([0, 1]) on the eigenvalues, we have [15]

$$\Sigma = \Lambda$$
$$Q = A = B.$$

From the above we have

$$\lim_{p \to n} B = Q \tag{17}$$

indicating the follow-up steps and final outputs would be exactly the same in two algorithms. ■

There are many choices for the kernel function (measuring the similarity between the data point $\mathbf{x}$ and the landmark $\mathbf{u}$) in (5). Several most popularly used schemes are listed as follows.

1) *0–1 (Binary):* $K(\mathbf{x}_i, \mathbf{u}_j) = 1$ if and only if the landmark $\mathbf{u}_j$ is among the $r$ nearest neighbors of $\mathbf{x}_i$ and $K(\mathbf{x}_i, \mathbf{u}_j) = 0$ otherwise. This is the simplest method.
2) *Gaussian (Heat Kernel):* If $\mathbf{u}_j$ is $\mathbf{x}_i$'s neighbor, define
$$K(\mathbf{x}_i, \mathbf{u}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{u}_j\|^2}{2h^2}}$$
which is chosen as our kernel function in the previous sections.
3) *Polynomial Kernel:* If $\mathbf{u}_j$ is $\mathbf{x}_i$'s neighbor, define
$$K(\mathbf{x}_i, \mathbf{u}_j) = (\mathbf{x}_i^T \mathbf{u}_j + 1)^d.$$
The parameter $d$ in the equation indicates the degree of the polynomial kernel. Order $d$ polynomial kernel can discover nonlinear structure with polynomial basis functions of order $d$.

TABLE II
DATA SETS USED IN OUR EXPERIMENTS

| Data set | # of instances | # of features | # of classes |
|---|---|---|---|
| MNIST | 70000 | 784 | 10 |
| LetterRec | 20000 | 16 | 26 |
| PenDigits | 10992 | 16 | 10 |
| Seismic | 98528 | 50 | 3 |
| Covtype | 581012 | 54 | 7 |

4) *Cosine:* If $\mathbf{u}_j$ is $\mathbf{x}_i$'s neighbor, define

$$K(\mathbf{x}_i, \mathbf{u}_j) = \mathbf{x}_i^T \mathbf{u}_j.$$

This scheme is usually applied when data points are normalized to have unit norm, which is commonly seen in document retrieval applications.

As Proposition 3 claims, no matter what schemes are selected in LSC, the asymptotic property is preserved. As an illustration, we will compare different kernels in the next section.

## V. EXPERIMENTS

In this section, several experiments were conducted to demonstrate the effectiveness of the proposed LSC.

### A. Data Sets

We have conducted experiments on five real-world large data sets downloaded from the UCI machine learning repository[2] and the LibSVM data sets page.[3] An brief description of the data sets is listed below (see Table II for some important statistics).

1) *MNIST:* A data set of handwritten digits from Yann LeCun's page.[4] The digits have been size-normalized and centered in a fixed-size image, and each image is represented as a 784 dimensional vector.
2) *LetterRec:* A data set of 26 capital letters in the English alphabet. 16 character image features are selected.
3) *PenDigits:* Also a handwritten digit data set of 250 samples from 44 writers, but it uses the sampled coordination information of each digit instead.
4) *Seismic:* A data set initially built for the task of classifying the types of moving vehicles in a distributed, wireless sensor network [10].
5) *Covtype:* A data set to predict forest cover type from cartographic variables of four wilderness areas located in the Roosevelt National Forest of Northern Colorado.

Each data point is normalized to have the unit norm and no other preprocessing step is applied.

### B. Evaluation Metric

The clustering result is evaluated by comparing the obtained label of each sample with the label provided by the data set. We use both the accuracy (AC) and normalized mutual information (NMI) metric [3] to measure the clustering performance. Given a data point $\mathbf{x}_i$, let $r_i$ and $s_i$ be the obtained cluster label and the label provided by the corpus, respectively. The AC is defined as follows:

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

where $n$ is the total number of samples and $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise, and $\mathrm{map}(r_i)$ is the permutation mapping function that maps each cluster label $r_i$ to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm [29].

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

$$\mathrm{MI}(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log \frac{p(c_i, c'_j)}{p(c_i) p(c'_j)}$$

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. $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected sample belongs to the clusters $c_i$ and $c'_j$ at the same time. In our experiments, we use the NMI as follows:

$$\mathrm{NMI}(C, C') = \frac{\mathrm{MI}(C, C')}{\max\left(H(C), H(C')\right)}$$

where $H(C)$ and $H(C')$ are the entropies of $C$ and $C'$. It is easy to check that both AC and NMI ranges from 0 to 1 and a higher value indicates a better clustering result.

We also record the running time of each method. All the codes in the experiments are implemented in MATLAB R2010a and run on a Linux machine with 2.66 GHz CPU, 4GB main memory.

### C. Compared Algorithms

To demonstrate the effectiveness and efficiency of our proposed LSC, we compare it with three other state-of-the-art approaches described in Section II. Following is a list of information concerning experimental settings of each method.

1) *KASP:* $k$-means-based approximate spectral clustering method proposed in [38]. The authors have provided their R code on the website.[5] For fair comparison, we implement a multiway partition version in MATLAB.
2) *CSC:* Committees-based spectral clustering proposed in [36]. Similar to that of KASP, we implement the code in MATLAB. We further set the threshold of points been replaced by committees to be 0.99, indicating only top 1% farthest ones are remained to make the acceleration more significant.
3) *Nyström:* There are several variants available for Nyström approximation-based spectral clustering, and we choose the MATLAB implementation with orthogonalization [5], which is available online.[6]

[2] http://archive.ics.uci.edu/ml
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
[4] http://yann.lecun.com/exdb/mnist/
[5] http://www.cs.berkeley.edu/~jordan/fasp.html
[6] http://alumni.cs.ucsb.edu/~wychen/

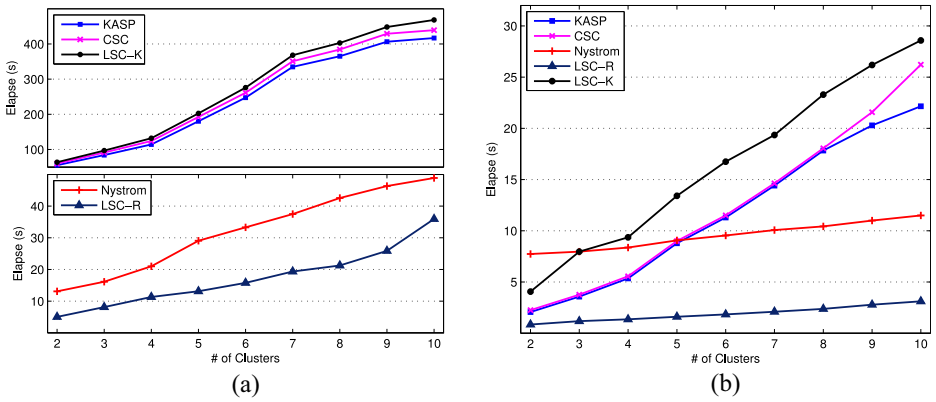


Fig. 1. Running time versus number of clusters on (a) MNIST and (b) PenDigits.

TABLE III
CLUSTERING TIME ON THE FIVE DATA SETS (S)

| Data set | Original | KASP | CSC | Nyström | LSC-R | LSC-K |
|---|---|---|---|---|---|---|
| MNIST | *3654.90* | 416.66 | 439.06 | 48.88 | **35.95** | 468.17 |
| LetterRec | *195.63* | 66.65 | 66.93 | 24.43 | **9.63** | 61.59 |
| PenDigits | *60.48* | 22.15 | 26.22 | 11.49 | **3.11** | 28.58 |
| Seismic | *4328.35* | **16.64** | 18.34 | 38.34 | 21.73 | 67.02 |
| Covtype | *181006.17* | 360.07 | 402.14 | 258.25 | **134.71** | 615.84 |

TABLE IV
CLUSTERING ACCURACY ON THE FIVE DATA SETS (%)

| Data set | Original | KASP | CSC | Nyström | LSC-R | LSC-K |
|---|---|---|---|---|---|---|
| MNIST | *72.46* | 56.51 | 55.51 | 53.70 | 62.66 | **67.04** |
| LetterRec | *31.04* | 29.49 | 27.12 | 30.11 | 29.22 | **30.33** |
| PenDigits | *76.55* | 72.47 | 70.78 | 73.94 | 79.04 | **79.27** |
| Seismic | *65.23* | 63.70 | 66.76 | 66.92 | 67.60 | **67.65** |
| Covtype | *44.24* | 22.42 | 21.65 | 22.31 | 24.75 | **25.50** |

To test the effectiveness of the accelerating scheme, we also report the results of the conventional spectral clustering. For our LSC, we implemented two versions as follows.

1) *LSC-R:* Short for landmark-based spectral clustering using random sampling to select landmarks.
2) *LSC-K:* Short for landmark-based spectral clustering using $k$-means for landmark-selection.

There are two parameters in our LSC approach: the number of landmarks $p$ and the number of nearest landmarks $r$ for a single point. Throughout our experiments, we empirically set $r = 6$ and $p = 500$.

For fair comparison, we use the same clustering result for landmarks (centers) selection in KASP, CSC, and LSC-K. We also use the same random selection for Nyström and LSC-R. For each landmark number $p$ (or number of centers, number of selected samples), 20 tests are conducted and the average performance is reported.

### D. Experimental Results

The performance of the five methods along with the original spectral clustering on all the five data sets are reported in Tables III–V.

TABLE V
NORMALIZED MUTUAL INFORMATION ON THE FIVE DATA SETS (%)

| Data set | Original | KASP | CSC | Nyström | LSC-R | LSC-K |
|---|---|---|---|---|---|---|
| MNIST | *75.22* | 55.58 | 54.67 | 48.02 | 61.42 | **68.30** |
| LetterRec | *41.05* | **40.16** | 37.14 | 39.12 | 37.34 | 39.63 |
| PenDigits | *74.04* | 68.13 | 66.93 | 66.81 | 74.94 | **76.24** |
| Seismic | *26.28* | **28.34** | 27.04 | 27.05 | 67.60 | 27.87 |
| Covtype | *20.39* | 7.44 | 7.19 | 7.42 | 8.31 | **9.02** |

In order to further examine the behaviors of these methods, we choose MNIST and PenDigits data sets and conducted a thorough study. Tables VI and VII and Fig. 1 show the clustering results on the MNIST and PenDigits data sets, respectively. We conduct the experiments with different clustering numbers in order to randomize the results. For each cluster number $k$, 20 test runs are performed on different randomly chosen clusters (except the case when the entire data set is used). These results reveals a number of interesting points as follows.

1) Considering the accuracy and NMI, LSC-K outperforms all of its competitors on all the data sets. For example, LSC-K achieves an 11% accuracy gain and 13% NMI gain on MNIST over the second best nonLSC method. It even beats the original spectral clustering algorithm on several data sets. The reason might be the effectiveness of the proposed landmark-based sparse representation. However, running time is its fatal weakness due to the $k$-means-based landmarks selection.
2) LSC-R demonstrates an elegant balance between running time and performance. It runs much faster than the other four methods while still achieves comparable accuracy and NMI with LSC-K. Particularly, on Covtype, it finishes in 135 s, which is almost 1500 times faster than the original spectral clustering. Comparing to LSC-K, LSC-R achieves a similar accuracy and NMI within 1/9 time on PenDigits. Overall, LSC-R is the best choice among the compared approaches.
3) The running time difference between LSC-R and LSC-K shows how the initial $k$-means performs. It is not surprising that the $k$-means-based landmark selection becomes

TABLE VI
CLUSTERING PERFORMANCE ON MNIST (%)

| k | Accuracy (%) | | | | | Normalized Mutual Information (%) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **KASP** | **CSC** | **Nyström** | **LSC-R** | **LSC-K** | **KASP** | **CSC** | **Nyström** | **LSC-R** | **LSC-K** |
| 2 | 84.85 | 83.85 | 88.20 | **96.42** | 96.30 | 52.73 | 52.50 | 55.42 | 86.54 | **87.80** |
| 3 | 81.29 | 81.48 | 73.48 | 86.59 | **94.09** | 58.44 | 59.23 | 46.73 | 71.46 | **84.90** |
| 4 | 66.48 | 67.67 | 66.32 | 77.38 | **81.34** | 48.45 | 49.62 | 47.15 | 68.65 | **74.05** |
| 5 | 60.35 | 60.05 | 61.40 | **70.75** | 69.89 | 49.05 | 48.87 | 46.96 | 68.01 | **70.11** |
| 6 | 62.65 | 62.27 | 59.08 | **71.58** | 71.03 | 53.49 | 54.32 | 47.37 | 67.49 | **71.87** |
| 7 | 61.85 | 61.87 | 60.37 | 68.17 | **73.38** | 56.49 | 55.97 | 49.39 | 65.26 | **74.50** |
| 8 | 59.27 | 59.93 | 57.00 | 64.52 | **69.70** | 56.21 | 56.67 | 49.55 | 63.40 | **71.41** |
| 9 | 56.20 | 56.92 | 54.52 | 63.28 | **68.25** | 54.61 | 54.52 | 48.92 | 61.48 | **69.06** |
| 10 | 56.51 | 55.51 | 53.70 | 62.66 | **67.04** | 55.58 | 54.67 | 48.02 | 61.42 | **68.30** |
| Avg. | 65.49 | 65.51 | 63.78 | 73.48 | **76.78** | 53.89 | 54.04 | 48.83 | 68.19 | **74.67** |

TABLE VII
CLUSTERING PERFORMANCE ON PENDIGITS (%)

| k | Accuracy (%) | | | | | Normalized Mutual Information (%) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **KASP** | **CSC** | **Nyström** | **LSC-R** | **LSC-K** | **KASP** | **CSC** | **Nyström** | **LSC-R** | **LSC-K** |
| 2 | 88.28 | **88.80** | 86.97 | 88.18 | 84.00 | 59.02 | 59.96 | 59.44 | **63.57** | 59.37 |
| 3 | 84.40 | 82.91 | 84.63 | **88.89** | 87.01 | 62.85 | 61.56 | 68.20 | **75.62** | 74.42 |
| 4 | 76.37 | 77.18 | 78.31 | **85.87** | 84.69 | 61.40 | 62.50 | 67.33 | **76.39** | 75.65 |
| 5 | 77.64 | 78.11 | 75.61 | 84.92 | **87.78** | 65.59 | 66.09 | 64.84 | 76.11 | **79.30** |
| 6 | 73.11 | 73.55 | 72.28 | 81.99 | **87.79** | 64.29 | 64.82 | 66.84 | 76.12 | **80.99** |
| 7 | 73.56 | 73.82 | 69.04 | 82.54 | **84.99** | 66.59 | 66.83 | 65.55 | 76.42 | **79.32** |
| 8 | 74.35 | 73.82 | 66.61 | 81.15 | **82.35** | 67.79 | 67.24 | 64.85 | 75.76 | **77.91** |
| 9 | 75.14 | 73.92 | 65.31 | 80.30 | **80.94** | 68.61 | 68.11 | 66.22 | 76.23 | **77.48** |
| 10 | 72.47 | 70.78 | 73.94 | 79.04 | **79.27** | 68.13 | 66.93 | 66.81 | 74.94 | **76.24** |
| Avg. | 77.26 | 76.99 | 74.74 | 83.65 | **84.31** | 64.92 | 64.89 | 65.56 | 74.57 | **75.63** |

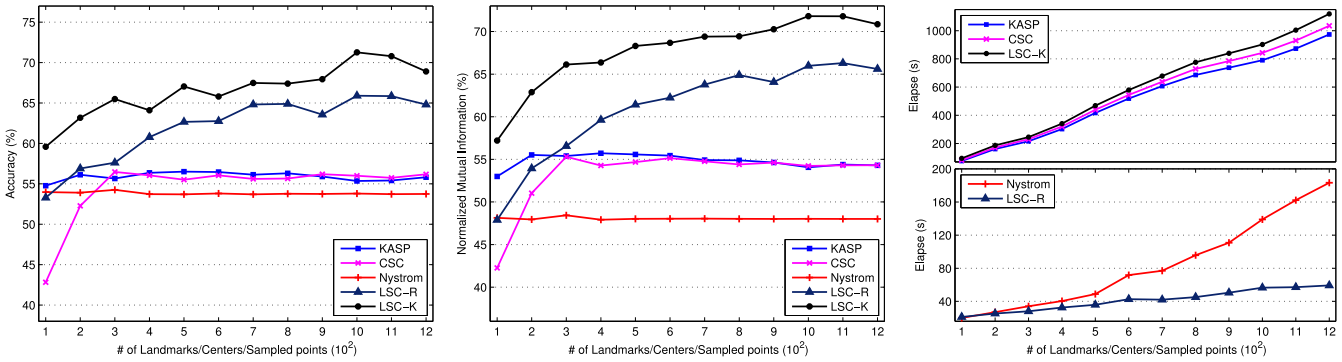


Fig. 2. Clustering accuracy, normalized mutual information, and running time versus number of landmark points on MNIST data set.

very slow as either the sample number or the feature number gets large.

4) As the cluster number $k$ gets larger, the advantage on the performance of LSC over other competing methods gets larger as well (as showed in Tables VI and VII). It implies that the LSC is capable to preserve the information of the data, even the cluster structure is complex.

### *E. Parameters Selection*

In previous experiments, all the compared algorithms share one parameter: the number of landmarks $p$ (the number of centers in KASP and CSC, and the number of sampled points in Nyström). Figs. 2 and 3 show how the clustering accuracy, NMI and running time changes as $p$ varying from 100 to 1200 on MNIST and PenDigits. It can be seen that LSC methods (both LSC-K and LSC-R) can achieve better clustering results as the number of landmarks increases.

Another essential parameter in LSC is the number of nearest landmarks $r$ for a single data point in sparse representation learning. Figs. 4 and 5 show how the clustering accuracy, NMI, and running time of LSC varies with this parameter. As we can see, LSC achieves consistent good performance with the $r$ varying from 3 to 10 despite a slight performance degradation

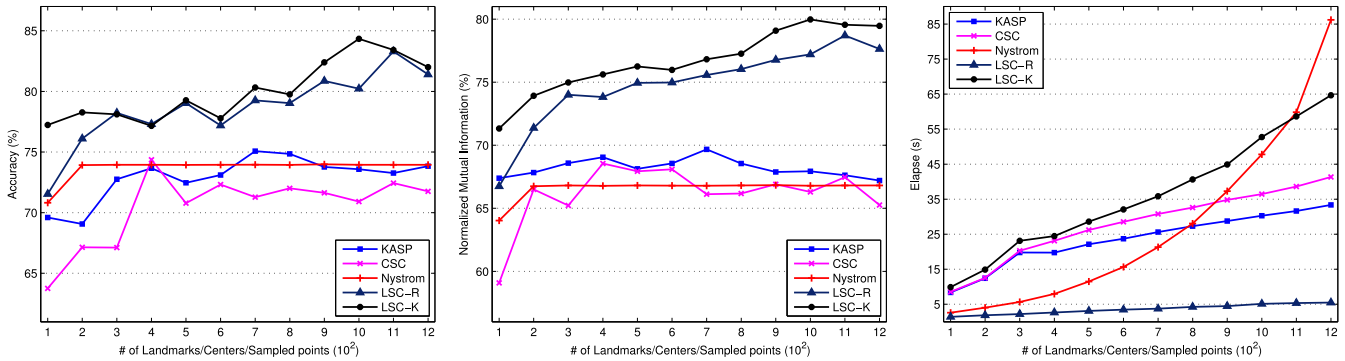


Fig. 3. Clustering accuracy, normalized mutual information, and running time versus number of landmark points on PenDigits data set.

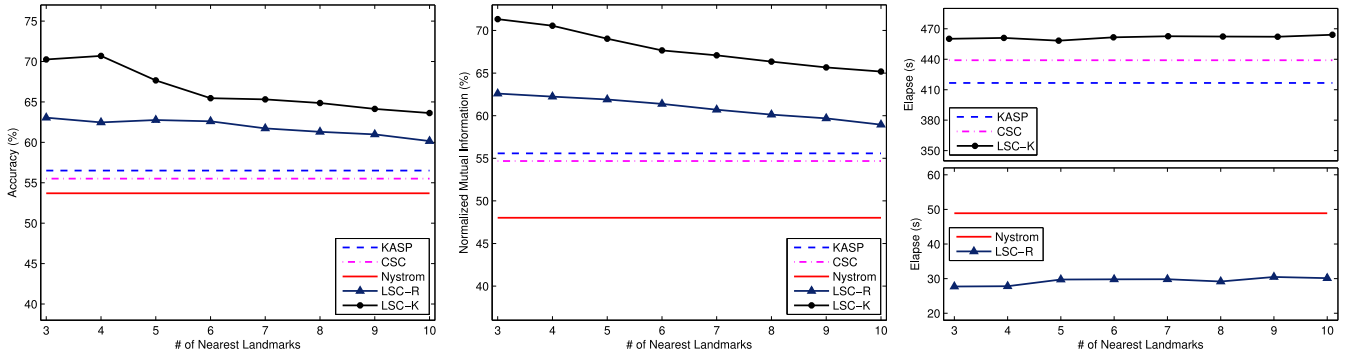


Fig. 4. Clustering accuracy, normalized mutual information, and running time versus number of nearest landmarks on MNIST data set.

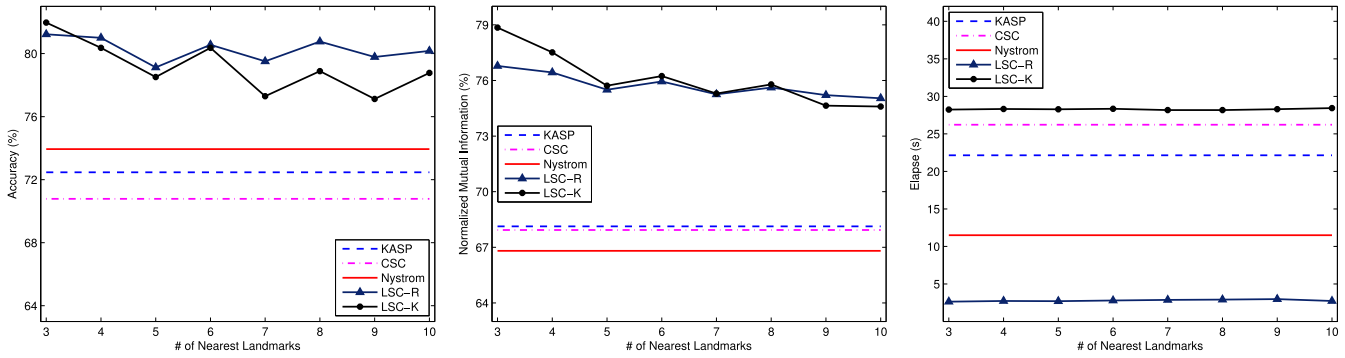


Fig. 5. Clustering accuracy, normalized mutual information, and running time versus number of nearest landmarks on PenDigits data set

as $r$ gets larger. This degradation complies to the theoretical analysis of MSE in (15), since the upper bound is quadratic to the distance of the farthest landmark $\epsilon_i$ to a point $\mathbf{x}_i$. It outweighs the impact of a larger $r$ in the denominator, which is proportional to $r$ itself. On MNIST data set, LSC-K out performs LSC-R, while on PenDigits data set, we can see random selection can achieve proximate, or even better result than the $k$-means selection with a much lower time span. This phenomenon, implies an overall robustness of the proposed landmark-base spectral clustering framework, regardless of the schemes of landmark selection.

### *F. Weighting Scheme Selection in Sparse Representation*

There are a number of choices of kernels (similarity measures) on constructing the sparse representation matrix $Z$ as listed in Section IV. For consistency, we use the heat kernel throughout our previous experiment. In this section, we would like to demonstrate how LSC works with other schemes.

We compare four kinds of weighting methods:

1) 0–1 weighting;
2) Gaussian kernel weighting, where the parameter $h$ is set as the average distance between two points in the data set;
3) cosine weighting;
4) polynomial kernel weighting with degrees 2 and 5 (denoted as "poly2" and "poly5").

As we can see from Table VIII, LSC is very robust with respect to weighting schemes, though performance is slightly varied, even we choose the most simple 0–1 weighting, we can achieve a similar result that stands out among the efficient spectral clustering methods compared.

TABLE VIII
COMPARISON USING DIFFERENT WEIGHTING FUNCTIONS ON MNIST(%)

| LSC-R | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| k | Accuracy (%) | | | | | Normalized Mutual Information (%) | | | | |
| | **0-1** | **Gaussian** | **Cosine** | **Poly2** | **Poly5** | **0-1** | **Gaussian** | **Cosine** | **Poly2** | **Poly5** |
| 2 | 96.42 | 96.42 | 96.42 | 96.42 | 96.42 | 86.55 | 86.54 | 86.55 | 86.55 | 86.56 |
| 3 | 86.58 | 86.59 | 86.58 | 86.58 | 86.59 | 71.45 | 71.46 | 71.45 | 71.45 | 71.48 |
| 4 | 77.38 | 77.38 | 77.38 | 77.38 | 77.39 | 68.65 | 68.65 | 68.65 | 68.65 | 68.66 |
| 5 | 70.75 | 70.75 | 70.75 | 70.75 | 70.76 | 68.01 | 68.01 | 68.01 | 68.01 | 68.02 |
| 6 | 71.58 | 71.58 | 71.58 | 71.58 | 71.59 | 67.49 | 67.49 | 67.49 | 67.49 | 67.51 |
| 7 | 68.12 | 68.17 | 68.54 | 67.62 | 68.56 | 65.24 | 65.26 | 65.31 | 65.03 | 65.33 |
| 8 | 63.91 | 64.52 | 64.50 | 64.30 | 64.41 | 63.09 | 63.40 | 63.39 | 63.25 | 63.38 |
| 9 | 63.27 | 63.28 | 63.16 | 63.40 | 63.40 | 61.41 | 61.48 | 61.40 | 61.48 | 61.60 |
| 10 | 62.37 | 62.66 | 62.26 | 62.38 | 61.91 | 61.23 | 61.42 | 61.46 | 61.45 | 60.98 |
| Avg. | 73.37 | 73.48 | 73.46 | 73.38 | 73.45 | 68.12 | 68.19 | 68.19 | 68.15 | 68.17 |
| LSC-K | | | | | | | | | | |
| k | Accuracy (%) | | | | | Normalized Mutual Information (%) | | | | |
| | **0-1** | **Gaussian** | **Cosine** | **Poly2** | **Poly5** | **0-1** | **Gaussian** | **Cosine** | **Poly2** | **Poly5** |
| 2 | 96.29 | 96.30 | 96.31 | 96.31 | 96.31 | 87.74 | 87.80 | 87.82 | 87.80 | 87.82 |
| 3 | 94.04 | 94.09 | 94.12 | 94.10 | 94.12 | 84.80 | 84.90 | 84.94 | 84.91 | 84.94 |
| 4 | 81.30 | 81.34 | 81.35 | 81.34 | 81.35 | 73.96 | 74.05 | 74.08 | 74.06 | 74.08 |
| 5 | 69.86 | 69.89 | 71.25 | 71.25 | 71.25 | 70.02 | 70.11 | 70.85 | 70.83 | 70.85 |
| 6 | 71.00 | 71.03 | 71.04 | 71.03 | 71.04 | 71.77 | 71.87 | 71.76 | 71.89 | 71.76 |
| 7 | 73.34 | 73.38 | 73.40 | 73.40 | 73.40 | 74.35 | 74.50 | 74.55 | 74.52 | 74.55 |
| 8 | 69.60 | 69.70 | 69.36 | 69.35 | 69.35 | 71.28 | 71.41 | 71.42 | 71.40 | 71.42 |
| 9 | 67.39 | 68.25 | 67.78 | 67.66 | 67.43 | 68.78 | 69.06 | 68.98 | 68.99 | 68.84 |
| 10 | 66.23 | 67.04 | 66.05 | 65.31 | 66.46 | 67.95 | 68.30 | 67.97 | 67.67 | 68.03 |
| Avg. | 76.56 | 76.78 | 76.74 | 76.64 | 76.75 | 74.52 | 74.67 | 74.71 | 74.68 | 74.70 |

## VI. CONCLUSION

In this paper, we have presented a novel large scale spectral clustering method, called LSC. Given a data set with $n$ data points, LSC selects $p$ ($\ll n$) representative data points as the landmarks and represent the original data points as the linear sparse combinations of these landmarks. The spectral embedding of the data can then be efficiently computed with the landmark-based representation. As a result, LSC scales linearly with the problem size. Extensive experiments on clustering show the effectiveness and efficiency of our approach comparing to the state-of-the-art methods.

Several questions remain to be investigated in our future work.

1) For the scalability goal, we do not really perform a sparse coding factorization, instead we turn to a heuristic technique based on the assumption that neighbor landmarks are more suitable to represent the data points. And in the experiment, we observe that this latent graph can achieve even better results than the original spectral clustering, which may imply a more powerful class of graphs. It is interesting to delve into the actual performance of applying sparse coding to graph construction.
2) There are two parameters $p$ and $\lambda$ to adjust in the algorithm. Given a data set of size $n \times m$, how to find out a suitable value of $p$ and $r$ is critical in the algorithm. It remains unclear how to do it theoretically and efficiently.
3) In this paper, we provided a general framework for LSC. We say it is general in that we can use different ways to generate landmarks. We have focused on $k$-means selection and random selection methods in this paper, and achieved promising results. However, according to our theoretical analysis, the problem of optimal landmarks selection can be formulated as an optimization problem. More elegant methods in this direction are to be discovered in the future.

## REFERENCES

[1] S. Balla-Arabe, X. Gao, and B. Wang, "A fast and robust level set method for image segmentation using fuzzy clustering and lattice Boltzmann method," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 910–920, Jun. 2013.
[2] C. Bishop, *Pattern Recognition and Machine Learning*, vol. 4. New York, NY, USA: Springer, 2006.
[3] D. Cai, X. He, J. Han, and S. Member, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
[4] B. Chen, B. Gao, T.-Y. Liu, Y.-F. Chen, and W.-Y. Ma, "Fast spectral clustering of data using sequential matrix compression," in *Proc. 17th Eur. Conf. Mach. Learn. (ECML)*, Berlin, Germany, 2006, pp. 590–597.
[5] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 568–586, Mar. 2010.
[6] F. R. K. Chung, *Spectral Graph Theory* (Regional Conference Series in Mathematics), vol. 92. Providence, RI, USA: AMS, 1997.
[7] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.
[8] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM Int. Conf. Knowl. Discov. Data Mining (SIGKDD)*, San Francisco, CA, USA, 2001, pp. 269–274.

[9] D. L. Donoho and M. Elad, "Optimally sparse representation in general (non-orthogonal) dictionaries via $l_1$ minimization," in *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 5, pp. 2197–2202, 2003.
[10] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 826–838, 2004.
[11] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.
[12] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *Proc. 2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 895–900.
[13] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognit.*, vol. 41, pp. 176–190, Jan. 2008.
[14] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.
[15] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.
[16] K. Gregor and Y. Lecun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 399–406.
[17] W. Härdle, *Applied Nonparametric Regression*. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[18] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *J. Roy. Statist. Soc. Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
[19] P. O. Hoyer and P. Dayan, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Nov. 2004.
[20] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
[21] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst. 19*, 2006, pp. 801–808.
[22] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area v2," in *Proc. Adv. Neural Inf. Process. Syst. 20*, 2008, pp. 873–880.
[23] M. Li, J. T. Kwok, and B. L. Lu, "Making large-scale Nyström approximation possible," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 631–638.
[24] P. Li, J. Bu, C. Chen, Z. He, and D. Cai, "Relational multimanifold coclustering," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1871–1881, Dec. 2013.
[25] H. Liu, G. Yang, Z. Wu, and D. Cai, "Constrained concept factorization for image representation," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1214–1224, Jul. 2014.
[26] T.-Y. Liu, H.-Y. Yang, X. Zheng, T. Qin, and W.-Y. Ma, "Fast large-scale spectral clustering by sequential shrinkage optimization," in *Proc. 29th Eur. Conf. Inf. Retriev. Res. (ECIR)*, Rome, Italy, 2007, pp. 319–330.
[27] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 679–686.
[28] Y. Liu *et al.*, "Understanding and enhancement of internal clustering validation measures," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 982–994, Jun. 2013.
[29] L. Lovasz and M. Plummer, *Matching Theory*. Budapest, Hungary: Akadémiai Kiadó, 1986.
[30] J. Mairal *et al.*, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.
[31] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst. 14*, 2001, pp. 849–856.
[32] 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, pp. 607–609, Jun. 1996.
[33] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1," *Vis. Res.*, vol. 37, pp. 3311–3325, Dec. 1997.
[34] T. Sakai and A. Imiya, "Fast spectral clustering with random projection and sampling," in *Proc. 3rd Int. Conf. Mach. Learn. Data Mining (MLDM)*, Leipzig, Germany, 2009, pp. 372–384.
[35] 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.
[36] H. Shinnou and M. Sasaki, "Spectral clustering for a large data set by reducing the similarity matrix size," in *Proc. 6th Int. Lang. Resources Evalu. (LREC)*, 2008, pp. 201–204.
[37] A. B. Tsybakov, *Introduction to Nonparametric Estimation*. 1st ed. New York, NY, USA: Springer, 2008.
[38] D. Yan, L. Huang, and M. I. Jordan, "Fast approximate spectral clustering," in *Proc. 15th ACM Int. Conf. Knowl. Discov. Data Mining (SIGKDD)*, Paris, France, 2009, pp. 907–916.
[39] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Bipartite graph partitioning and data clustering," in *Proc. 10th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2001, pp. 25–32.
[40] K. Zhang, I. W. Tsang, and J. T. Kwok, "Improved Nyström low-rank approximation and error analysis," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 1232–1239.
[41] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, 2006.

**Deng Cai** (M'09) received the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2009.

He is a Professor with the State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, China. His current research interests include machine learning, data mining, and information retrieval.

**Xinlei Chen** (S'12) received the B.S. degree in computer science from the Zhejiang University of China, Hangzhou, China, in 2012. He is currently pursuing the Ph.D. degree from the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.

His current research interests include large-scale problems in natural language processing and machine learning.