An Out-of-sample Extension of Sparse Subspace Clustering and Low Rank Representation for Clustering Large Scale Data Sets

Xi Peng, *Student Member, IEEE,* Lei Zhang, *Member, IEEE,* Zhang Yi, *Senior Member, IEEE*

I. Introduction

CLUSTERING algorithm is one of fundamental topics in pattern recognition and data mining, which aims to group the similar patterns into the same cluster by maximizing the inter-cluster dissimilarity and the intra-cluster similarity. Over the past two decades, a number of clustering approaches such as k-means algorithm [1], [2] have been extensively studied. Recently, clustering in linearly inseparable data has become a hot topic, and numerous methods have been proposed. For example, kernel-based clustering [3] and spectral clustering [4]. In this paper, we focus on the stability issue in spectral clustering algorithms.

Spectral clustering belongs to the family of subspace clustering (subspace segmentation) [5], [6] which aims at finding a low-dimensional subspace for each group of points. A widely-used assumption in subspace clustering is that the high-dimensional data actually locate at a low-dimensional subspace. Thus, the data could be grouped in a projected space. Thus, subspace clustering mainly contains two tasks, projecting the data set into an latent space (i.e., encoding) and calculating the cluster membership of the data set in the space (i.e., clustering). Some subspace clustering algorithms have been proposed and applied into data clustering and image processing, e.g., Density-based approach [7], algebraic algorithms [8] and statistical methods [9], [10].

Spectral clustering derives from the pioneer works in manifold learning [11]. Its basic idea is to find a cluster membership of the data points by using the spectrum of the affinity matrix that depicts the similarity among data points. Thus, the construction of similarity graph lies into the heart of spectral clustering algorithms. In a similarity graph, each vertex denotes a data point and the connection weight between two points represents their similarity.

Generally, there are two kinds of metrics to build a similarity graph, i.e., pairwise distance and reconstruction coefficients. The first metric measures the similarity by computing the distance between two data points, e.g., Euclidean distance. Pairwise distance could capture the local structure of data set, whose value only depends on the distance between two data points. As a result, it is sensitive to noises and outliers. Alternatively, representation coefficients-based methods assume that each data point can be denoted as a linear combination of other data points, and thus the representation coefficient can be regarded as a kind of measurement. The metric is robust against noise and outliers since the value of coefficient not only depends on the two connected data points, but also depends on the other data points. In other words, reconstruction coefficients is data-adaptive. Several recent works have shown that representation coefficient is superior to pairwise distance-based similarity in subspace clustering. For example, ℓ_1-norm-based sparse representation [12], [13], [14], the lowest-rank representation [15], the latent low rank representation [16], fixed rank representation [17], and linear regression-based methods [18], [19].

Although representation-based subspace clustering algorithms have achieved state-of-the-art performance in data clustering, these methods have suffered from scalability issue and out-of-sample problem. In this paper, we propose a generalized framework to resolve these problems and apply the framework to two most popular approaches, Sparse Subspace Clustering (SSC) [12], [13] and Low Rank Representation (LRR) [15]. SSC constructs a sparse similarity graph by using ℓ_1-minimization-based coefficients, whereas LRR uses the lowest-rank representation to depict the similarity among data points. Both SSC and LRR could automatically select the nearby points for each datum without fixing the size of neighborhood. However, SSC must solve n optimization problems involving in n data points and calculate the eigenvectors of a $n \times n$ graph Laplacian matrix. Its computational complexity is more than $O(n^3)$ even though the fastest ℓ_1-solver is used. This means that any medium-sized data set will bring up the scalability issues with SSC. Moreover, SSC does not handle the data (out-of-sample data) that are not used to construct the similarity graph. For each new datum, SSC has to perform the algorithm over the whole data set to obtain the similarity graph and cluster membership. This makes SSC is not suitable for fast online clustering. On the other hand, LRR suffers from the same problems as well, whose time complexity is proportion to the cube of the problem size.

In this paper, we propose a simple but effective method which makes SSC and LRR feasible to clustering out-of-sample data and large scale data. The work is based on the two-dimensional sparsity assumption that each subspace is low-dimensional and each data point can be encoded as a linear combination of a few basis vectors. Thus, the union of the subspaces spanned by basis vectors could equal or approximate to that spanned by the original data. In other words, one

Xi Peng, Lei Zhang and Zhang Yi are with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu 610065, P.R. China. (E-mail: pangsaai@gmail.com; leizhang@scu.edu.cn; zhangyi@scu.edu.cn).

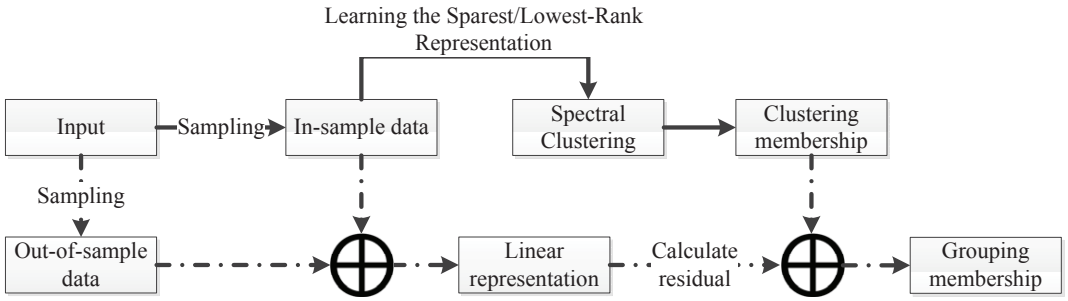


Fig. 1. Architecture of the proposed framework for scalable subspace clustering. Two kinds of lines are used to show the processes of clustering of in-sample data and out-of-sample data.

could use a small number of data points (in-sample data) to represent the original subspaces without loss of information. In theory, the solving of scalability issue is not at the cost of clustering quality if the basis vectors are sampled. We believe that this characteristic may only belong to the representation coefficient-based subspace clustering approaches and has not been exploited to develop a scalable method as far as we known.

The proposed method resolves the scalability issue and out-of-sample problem in SSC and LRR in the manner of "sampling, clustering, coding, and classifying" (Fig. 1). In details, we split the data into two parts, in-sample data and out-of-sample data, where out-of-sample data are drawn from the subspaces spanned by in-sample data; and obtain the cluster assignment matrix of in-sample data by performing SSC or LRR; after that, assign each out-of-sample datum to the nearest subspace that has minimal residual.

Several perspectives might be highlighted to our work:

1) The proposed method is a generalized framework which makes representation-based subspace clustering algorithms feasible in large scale setting. It groups the new coming samples without recalculating the reconstruction coefficients and cluster membership of the whole data set. This is beneficial to fast online clustering.
2) Following our framework, we present an out-of-sample extension of SSC and LRR, i.e., Scalable Sparse Subspace Clustering (SSSC) and Scalable Low Rank Representation (SLRR). Both SSSC and SLRR reduce the time complexity of the original algorithms from the cube to linearity of the problem size, while still preserves good performance. Extensive experiments show that SSSC and SLRR achieved state-of-the-art results in scalable data clustering.
3) When the subspaces are independent, we proved that SLRR can exactly reveal the membership of the data points even though the data set contains outliers. In other words, we resolved the scalability issue and out-of-sample problem in SSC and LRR without the loss in clustering quality when some conditions are satisfied.

The rest of the paper is organized as follows: Section II provides a brief review of SSC, LRR and some scalable spectral clustering algorithms. Section III presents our framework and applies it to two popular representation-based spectral clustering algorithms (SSC and LRR). Section IV carries out the experiments to examine the effectiveness of our methods in large scale setting. Finally, Section V concludes this work.

TABLE I
NOTATIONS.

Notation	Definition
n	the number of data points
m	the dimensionality of data
k	the desired number of clusters
p	the number of in-sample data
t	the number of iterations of algorithm
$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$	data set
$\mathbf{Y}_i = \mathbf{Y} \setminus \mathbf{y}_i$	the data set $\mathbf{Y}$ except $\mathbf{y}_i$
$[\mathbf{Y}]_i$	the data points in the ith subspace
$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]$	the representation coefficients of $\mathbf{Y}$
$\mathbf{A} \in \mathbb{R}^{n \times n}$	affinity matrix
$\mathbf{L} \in \mathbb{R}^{n \times n}$	Laplacian matrix
$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$	the first k eigenvectors of $\mathbf{L}$
$\mathbf{V} \in \mathbb{R}^{n \times k}$	eigenvector matrix
$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p]$	in-sample data
$\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_{n-p}]$	out-of-sample data

II. RELATED WORKS

Except in some specified cases, **lower-case bold letter** represents a column vector and **upper-case bold one** represents a matrix. $\mathbf{A}^T$ denotes the transpose of the matrix $\mathbf{A}$ whose pseudo-inverse is $\mathbf{A}^{-1}$, and $\mathbf{I}$ denotes identity matrix. Moreover, we summarize some notations used throughout the paper in TABLE I.

A. Sparse Subspace Clustering

Recently, some researchers have utilized sparse representation for various tasks, e.g., face recognition [20]. In these works, Elhamifar and Vidal [12], [13] proposed the SSC algorithm for subspace segmentation with well-founded recovery theory for independent subspaces and disjoint subspaces. SSC calculates the similarity among data points by solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{C},\mathbf{E},\mathbf{Z}} \|\mathbf{C}\|_1 + \lambda_E \|\mathbf{E}\|_1 + \lambda_Z \|\mathbf{Z}\|_F \\ &\text{s.t. } \mathbf{Y} = \mathbf{Y}\mathbf{C} + \mathbf{E} + \mathbf{Z}, \operatorname{diag}(\mathbf{C}) = 0, \end{aligned} \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{n \times n}$ is the sparse representation of the data set $\mathbf{Y} \in \mathbb{R}^{m \times n}$, $\mathbf{E}$ corresponds to the sparse outlying entries,

$\mathbf{Z}$ denotes the reconstruction errors owing to the limited representational capability, and the parameters λ_E and λ_Z balance the three terms in the objective function.

When the data set not contains outliers or the outlying entries have not sparse structure, (1) could be rewritten as follows

$$\min_{\mathbf{C}} \|\mathbf{C}\|_1 + \lambda\|\mathbf{Y} - \mathbf{Y}\mathbf{C}\|_F \quad \text{s.t. } \mathrm{diag}(\mathbf{C}) = 0, \tag{2}$$

or equivalently,

$$\min \ \|\mathbf{c}_i\|_1 \qquad \text{s.t. } \ \|\mathbf{y}_i - \mathbf{Y}_i\mathbf{c}_i\|_2 < \delta, \tag{3}$$

where $\mathbf{c}_i \in \mathbb{R}^n$ is the sparse representation of the data point $\mathbf{y}_i \in \mathbb{R}^m$ over the dictionary $\mathbf{Y}_i \triangleq [\mathbf{y}_1 \dots \mathbf{y}_{i-1}\ \mathbf{0}\ \mathbf{y}_{i+1} \dots \mathbf{y}_n]$, and $\delta \geq 0$ is the error tolerance. The above problems can be solved by a lot of convex optimization algorithms referring to [21] for an extensive survey.

After getting the sparsest coefficients of all data points, SSC performs spectral clustering [4] over the coefficients as described in Algorithm 1.

Algorithm 1 Sparse Subspace Clustering (SSC).

Input: A set of data points $\mathbf{Y} \in \mathbb{R}^{m\times n}$, and the desired number of clusters k.

1: Solve the ℓ_1-minimization problem (1), (2) or (3) to get $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n]$.
2: Form an affinity matrix $\mathbf{A} = |\mathbf{C}|^T + |\mathbf{C}|$.
3: Construct a Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ using $\mathbf{A}$, where $\mathbf{D} = \mathrm{diag}\{d_i\}$ with $d_i = \sum_{j=1}^{n} \mathbf{A}_{ij}$.
4: Obtain the eigenvector matrix $\mathbf{V} \in \mathbb{R}^{n\times k}$ which consists of the first k normalized eigenvectors of $\mathbf{L}$ corresponding to its k smallest eigenvalues.
5: Get the segmentations of the data by performing k-means on the rows of $\mathbf{V}$.

Output: The cluster assignments of $\mathbf{Y}$.

It is easy to find that the computational complexity of SSC is very high. Specifically, SSC needs $O(tn^2m^2 + tmn^3)$ to construct a similarity graph even though it adopts Homotopy optimizer [22] to get the sparsest solution, where Homotopy optimizer is one of the fastest ℓ_1-minimization algorithm according to [21] and t denotes the number of iterations of the optimizer. In addition, it will take $O(n^3)$ to calculate the eigenvectors of the Laplacian matrix $\mathbf{L}$. Considering $\mathbf{L}$ is a sparse matrix, the time complexity of this step could be reduced to $O(mn + mn^2)$ when Lanczos eigensolver is used. However, it is still a daunting task even for a moderate $n > 100,000$.

B. Low Rank Representation for Subspace Clustering

Exploring in the structure of data space is a challenging task in diverse set of fields, which often relates to a rank-minimization problem. One of extensively studied rank-minimization models is:

$$\min \ rank(\mathbf{C}) \quad \text{s.t. } \mathbf{Y} = \mathcal{A}(\mathbf{C}) + \mathbf{E}, \tag{4}$$

where $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m'\times n'}$ is a linear operator and $\mathbf{E}$ is an unknown matrix to learn.

However, problem (4) is a NP-hard problem for which all known finite time algorithms have at least doubly exponential running times in both theory and practice. Benefit from the developments in compressive sampling theory [23], one could get an approximate solution of (4) via

$$\min \ \|\mathbf{C}\|_* \qquad \text{s.t. } \mathbf{Y} = \mathcal{A}(\mathbf{C}) + \mathbf{E}, \tag{5}$$

where $\|\cdot\|_*$ denotes the nuclear norm, the sum of the singular values.

The above formulation mainly contains three variants for different problems, i.e., matrix completion [24], robust principal component analysis [25], and Low Rank Representation (LRR) [15], [16], [26], [27]. The differences in the algorithms are the choices of the linear operator $\mathcal{A}$ and the constraint over $\mathbf{E}$. LRR computes the lowest-rank representation over the data set itself and assumes $\mathbf{E}$ is the corruption with limited nonzero entries. Mathematically, it aims to solve

$$\min \ \|\mathbf{C}\|_* + \lambda\|\mathbf{E}\|_\ell \quad \text{s.t. } \mathbf{Y} = \mathbf{Y}\mathbf{C} + \mathbf{E}, \tag{6}$$

where $\|\cdot\|_\ell$ could be chosen as $\ell_{2,1}$-norm, ℓ_1-norm, or Frobenius norm. The choice only depends on which kind of error is assumed in the data set. In details, $\ell_{2,1}$-norm is usually adopted to depict sample-specific corruptions and outliers, ℓ_1-norm is more suitable to characterize random corruption, and Frobenius norm is used to describe small Gaussian noise.

To obtain the solution of the problem (6), Liu et al. [15] adopted Augmented Lagrange Multipliers (ALM) method to solve nuclear norm regularized optimization problems. Generally, their algorithm takes $O(m^2n + n^3)$ at each iteration to perform SVD over a dense matrix. In addition, it will take $O(n^3)$ to calculate the eigenvectors of the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n\times n}$ and $O(t_2nk^2)$ to perform k-means over the eigenvector matrix of $\mathbf{L}$, where t_2 denotes the number of iterations of k-means. Therefore, the overall time complexity of LRR is $O(t_1(m^2n+n^3)+t_2nk^2)$, where t_1 is the number of iterations of ALM. This complexity makes LRR inefficient in large scale setting. In Algorithm 2, we outline the implementation of LRR.

Furthermore, SSC and LRR cannot cope with the data that not used to construct the affinity matrix, i.e., they suffer from out-of-sample problem. For each new coming datum, SSC and LRR have to recalculate the sparsest/lowest-rank representation and the cluster assignment matrix of the whole data set. This makes SSC and LRR impossible to fast online clustering tasks.

C. Scalable Spectral Clustering Algorithm

In recent years, some works have devoted to solve the scalability issue in spectral clustering. One natural option is to reduce the time cost of eigen-decomposition over Laplacian matrix. Fowlkes et al. [28] proposed using Nyström method to avoid computing the whole similarity matrix. Li et al. [29] performed randomized low-rank matrix approximation algorithm over a subset of the input to find the representative data points (landmarks or exemplars). As a result, the Nyström method could be more efficient and effective in large scale setting.

Algorithm 2 Low Rank Representation for Subspace Clustering (LRR).

Input: A set of data points $\mathbf{Y} \in \mathbb{R}^{m\times n}$, balanced factor $\lambda > 0$, and the desired number of clusters k.

1: Get the lowest-rank representation coefficients $\mathbf{C} \in \mathbb{R}^{n\times n}$ by solving

$$\min \ \|\mathbf{C}\|_* + \lambda\|\mathbf{E}\|_\ell \quad \text{s.t. } \mathbf{Y} = \mathbf{YC} + \mathbf{E}$$

2: Form an affinity matrix $\mathbf{A} = |\mathbf{C}|^T + |\mathbf{C}|$.

3: Construct a Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ using $\mathbf{A}$, where $\mathbf{D} = \text{diag}\{d_i\}$ with $d_i = \sum_{j=1}^{n} \mathbf{A}_{ij}$.

4: Obtain the eigenvector matrix $\mathbf{V} \in \mathbb{R}^{n\times k}$ which consists of the first k normalized eigenvectors of $\mathbf{L}$ corresponding to its k smallest eigenvalues.

5: Get the segmentations of the data by performing k-means over the rows of $\mathbf{V}$.

Output: The cluster assignment of $\mathbf{Y}$.

Moreover, Chen et al. [30] performed eigen-decomposition in a distributed systems.

Another option is to reduce the data size by performing sampling techniques or replacing the original data set with a small number of points. Yan et al. [31] presented a method for fast approximate spectral clustering by selecting some representative points based on k-means or random projection trees. Chen and Cai [32] proposed an approach, called Landmark-based Spectral Clustering (LSC). It firstly chooses p representative points from data using k-means clustering or randomly sampling; and constructs a Laplacian matrix $\mathbf{L} = \mathbf{A}^T\mathbf{A}$, where the element $\mathbf{a}_{ij}$ of $\mathbf{A} \in \mathbb{R}^{p\times n}$ is the pairwise distance between the original data point and the landmarks. finally, performs spectral clustering over $\mathbf{L}$. Wang et al. [33] select landmarks by performing selective sampling technique; and then perform spectral clustering over the chosen samples based on pairwise distance; after that, project non-sampled data into a low-dimensional space using locality preserving projections algorithm [34]; finally, get the labels of the non-chosen data by using k-nearest neighbor classifier in the embedding space. Furthermore, Nie et al. [35] proposed Spectral Embedded Clustering (SEC) which groups new coming data by performing subspace learning method and the nearest classifier.

The second option, which selects some key data points to represent the others, has became more and more popular owing to its effectiveness and efficiency. However, all these approaches mainly focus on speeding up the classical spectral clustering methods (pairwise similarity-based ones) and did not explore the intrinsic characteristics of data structure. To the best our knowledge, only Peng et al. [36] developed an scalable version of SSC. Moreover, Lin et al. [37] proposed an optimization algorithm to linearize the quadratic penalty term of LRR and allow the penalty to change adaptively. As a result, the time complexity to learn the lowest-rank representation is reduced to $O(trn^2)$. Liu and Yan [38] presented an approach to reduce the time cost for learning the lowest-rank matrix from $O(t(m^2n + n^3))$ to $O(tmnr)$, where t is the number of iterations of the optimization algorithm, m is the dimensionality of the input, n denotes the number of data points, and r is the rank of the data matrix. Moreover, Zhang et al. [39] performed locality sensitive hashing over the truncated LRR [26] to build a sparse affinity graph such that the time complexity of learning representation linearly scales with the problem size. However, these works mainly focused on solving scalability issue in representation learning rather than developing an scalable representation-based subspace clustering method, i.e., these methods still suffer from out-of-sample problem and scalability issue in the context of subspace clustering.

III. The Proposed Algorithms

In this section, we present our framework which makes the representation-based subspace clustering feasible to clustering large scale data sets and new samples. To verify the efficacy of the framework, we apply it to SSC [12], [13] and LRR [15] and obtain the corresponding algorithms, named Scalable Sparse Subspace Clustering (SSSC) and Scalable Low Rank Representation (SLLR). SSSC and SLLR treat the scalability issue in SSC and LRR as an out-of-sample problem. They resolve the problems in the manner of "sampling, clustering, coding, and classifying". The first two steps choose a small number of data points as in-sample data and calculate the cluster membership of them. The third and fourth steps find a low-dimensional subspace to fit each out-of-sample data point and assign these data points to the subspace that have the minimal residual.

A. Algorithm Description

The basic idea of our approach is: for a given set of data points $\mathbf{Y}$ drawn from an union of linear subspaces $\{S_i\}_{i=1}^k$, one could use a small number of data points to represent the original data space. It is based on the following assumption:

Assumption 1 (two-dimensional sparsity): For the subspace S_i spanned by the data points $[\mathbf{Y}]_i \in \mathbb{R}^{m\times n_i}$, one could use a small portion of $[\mathbf{Y}]_i$, denoted as $\mathbf{X}_i \in \mathbb{R}^{m\times p_i}$, to represent S_i, where $rank(\mathbf{X}_i) = rank([\mathbf{Y}]_i)$ and $rank([\mathbf{Y}]_i) \leq p_i \ll n_i$.

Assumption 1 is reasonable and has been adopted in numerous works [34], [35], [40], which contains one-dimensional sparsity assumption that each data point could be encoded by a small number of data points. In fact, these two assumptions are two sides of the same coin. Moreover, Assumption 1 implies that out-of-sample data have no influence on the clustering result and the possibility that the scalability issue in SSC and LRR would be resolved without loss of clustering quality. In other words, SSSC (SLRR) will achieve comparable clustering quality with SSC (LRR) if the basis vectors are sampled, while the computational complexity is reduced from the cube to linearity of the problem size.

From the above analysis, we can see that it is a key to identify the representative points as in-sample data such that the original subspaces could be approximately or exactly spanned by the sampled data points (i.e., in-sample data). To address

this problem, some approaches have been proposed [41]. However, the computational complexity of these methods is very high such that it is impractical to apply them to handle large scale data set. Thus, we get in-sample data by adopting uniform random sampling technique whose time complexity is only $O(1)$. Furthermore, uniform random sampling method has been proved effective in large scale setting [28], [31], [32], [42].

After performing uniform random sampling, we calculate the clustering membership of in-sample data by using SSC or LRR. As the original subspaces could be approximately or exactly spanned by in-sample data, the loss of clustering quality of our method mainly comes from the grouping error of out-of-sample data. Therefore, it is a key to design an effective way to group these non-sampled data. The most simple method is to directly assign out-of-sample data to the nearest subspace in terms of Euclidean distance or other pairwise distances. However, most high-dimensional data are not lie into the Euclidean space such that Euclidean distance is not a good metric to measure the adjacency relationship among data points. On the other hand, a important task of subspace clustering is to find a low-dimensional representation for each data point. To solve these two problems, we compute the sparse representation of non-sampled data by using in-sample data as dictionary, and then assign each data point to the nearest subspace by performing Sparse Representation-based Classification (SRC) [20]. In details, for each out-of-sample data point $\bar{\mathbf{x}}_i \in \bar{\mathbf{X}}$, the following optimization problem is solved by

$$\min \ \|\bar{\mathbf{c}}_i\|_1 \quad \text{s.t.} \ \|\bar{\mathbf{x}}_i - \mathbf{X}\bar{\mathbf{c}}_i\|_2 < \delta, \tag{7}$$

where $\delta > 0$ is the error tolerance and $\mathbf{X}$ is in-sample data.

Once the optimal $\bar{\mathbf{c}}_i$ is achieved, $\bar{\mathbf{x}}_i$ is assigned to the nearest subspace that has the minimum residual via

$$r_j(\bar{\mathbf{x}}_i) = \|\bar{\mathbf{x}}_i - \mathbf{X}\delta_j(\bar{\mathbf{c}}_i)\|_2. \tag{8}$$

$$\text{identity}(\bar{\mathbf{x}}_i) = \underset{j}{\text{argmin}}\{r_j(\bar{\mathbf{x}}_i)\}, \tag{9}$$

where the nonzero entries of $\delta_j(\bar{\mathbf{c}}_i) \in \mathbb{R}^n$ are the elements in $\bar{\mathbf{c}}_i$ that are associated with the j-th subspace, and the $\text{identity}(\bar{\mathbf{x}}_i)$ denotes the assignment of $\bar{\mathbf{x}}_i$.

Although SRC has achieved impressive results in pattern recognition, some recent works [43], [44] empirically showed that non-sparse linear representation could achieve competitive recognition rate with less time cost. Therefore, to reduce the computing cost for grouping out-of-sample data, the main block in large scale clustering, we perform linear coding scheme but sparse one by solving

$$\min_{\bar{\mathbf{c}}_i} \|\bar{\mathbf{x}}_i - \mathbf{X}\bar{\mathbf{c}}_i\|_2^2 + \gamma\|\bar{\mathbf{c}}_i\|_2^2, \tag{10}$$

where $\gamma \ll 1$ is a positive real number. The second term is called ridge regression in statistics, which is used to avoid over-fitting. Zhang et al. [43] named the solution of (10) as collaborative representation and showed that collaborative representation but sparse representation plays an important role in face recognition.

Algorithm 3 Scalable Sparse Subspace Clustering (SSSC) and Scalable Low Rank Representation (SLRR).

Input: A set of data points $\mathbf{Y} \in \mathbb{R}^{m\times n}$, the desired number of clusters k, and the rigid regression parameter $\gamma = e^{-6}$.

1: Randomly select p data points from $\mathbf{Y}$, denoted by $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p)$, as in-sample data. The remaining samples are used as out-of-sample data, denoted by $\bar{\mathbf{X}} = (\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_{n-p})$.

2: Perform SSC (Algorithm 1) or LRR (Algorithm 2) over $\mathbf{X}$ to get the cluster membership of $\mathbf{X}$.

3: Calculate the linear representation of out-of-sample data point $\bar{\mathbf{x}}_i$ by using $\mathbf{X}$ as dictionary, i.e.,

$$\bar{\mathbf{c}}_i^* = (\mathbf{X}^T\mathbf{X} + \gamma\mathbf{I})^{-1}\mathbf{X}^T\bar{\mathbf{x}}_i.$$

4: Calculate the regularized residuals of $\bar{\mathbf{x}}$ over all subspaces via solving

$$r_j(\bar{\mathbf{x}}_i) = \frac{\|\bar{\mathbf{x}}_i - \mathbf{X}\delta_i(\mathbf{c}_i^*)\|_2}{\|\delta_j(\mathbf{c}_i^*)\|_2}.$$

5: Assign $\bar{\mathbf{x}}_i$ to the subspace which produces the minimal residual by

$$\text{identity}(\bar{\mathbf{x}}_i) = \underset{j}{\text{argmin}}\{r_j(\bar{\mathbf{x}}_i)\}.$$

Output: The cluster membership of $\mathbf{Y}$.

Then, each $\bar{\mathbf{x}}_i$ is assigned to the nearest subspace by calculating regularized residuals over all classes by

$$r_j(\bar{\mathbf{x}}_i) = \frac{\|\bar{\mathbf{x}}_i - \mathbf{X}\delta_j(\bar{\mathbf{c}}_i)\|_2}{\|\delta_j(\bar{\mathbf{c}}_i)\|_2}, \tag{11}$$

and assigning $\bar{\mathbf{x}}_i$ to the subspace which produces the minimal $r_j(\bar{\mathbf{x}}_i)$ by

$$\text{identity}(\bar{\mathbf{x}}_i) = \underset{j}{\text{argmin}}\{r_j(\bar{\mathbf{x}}_i)\}, \tag{12}$$

Algorithm 3 summarizes our algorithms which make SSC and LRR feasible to coping with out-of-sample data and large scale data set.

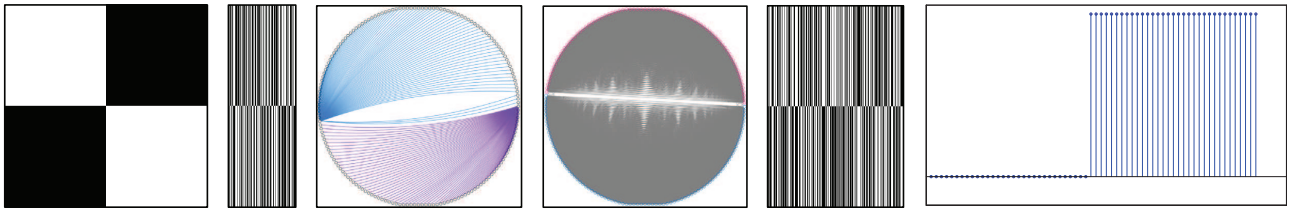
B. Theoretical Analysis for Independent Subspaces

In this subsection, we show that SLRR can exactly reveal the cluster membership of the data points when the subspaces are independent. As [15] did, we assume that the data are noise free or a small fraction of data samples are corrupted (so-called sample-specific corruptions[1]).

The grouping error of SLRR derives from two parts, one is performing LRR over in-sample data; the other is grouping out-of-sample data by minimizing residual over different subspaces. In the following, we prove that SLRR can exactly segment the data into multiple subspaces.

Theorem 1 ([15]): There exists $\epsilon^* > 0$ such that LRR with parameter $\lambda = 3/(7\|\mathbf{X}\|\sqrt{\epsilon^* p})$ strictly succeeds, as long as

[1]Outliers could be regarded as a kind of sample-specific corruptions.



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

Fig. 2. A toy example for showing the effectiveness of SSSC and SLRR. (a) Input data; (b) In-sample data; (c) The similarity graph of in-sample data achieved by SSSC; (d) The similarity graph of in-sample data achieved by SLRR; (e) Out-of-sample data; (f) The linear coefficient of a non-sampled data point.

$\epsilon \leq \epsilon^*$. Here, the success is in a sense that any minimizer $(\mathbf{C}^*, \mathbf{E}^*)$ to

$$\min \ \|\mathbf{C}\|_* + \lambda\|\mathbf{E}\|_{2,1} \qquad \text{s.t. } \mathbf{Y} = \mathbf{YC} + \mathbf{E}, \tag{13}$$

can produce

$$\mathbf{U}^*(\mathbf{U}^*)^T = \mathbf{V}_0^*(\mathbf{V}_0^*)^T \text{ and } \Gamma^* = \Gamma_0, \tag{14}$$

where ϵ defines the fraction of outliers or corruption, $\mathbf{U}^*$ is the column space of $\mathbf{C}^*$, and Γ^* is column support of $\mathbf{E}^*$. Note that, $\mathbf{E}$ is assumed having sparse column support and the subspaces are linearly independent.

Theorem 1 shows that SLRR exactly segments in-sample data points into multiple subspaces and identifies the corrupted samples. Thus, it is easy to remove the corruptions from in-sample data to obtain a clean dictionary for out-of-sample data.

Theorem 2: Consider a collection of data points $\mathbf{X}$ drawn from k linear independent subspaces $\{S_i\}_{i=1}^k$, let $[\mathbf{X}]_i$ denote the data points in S_i and $[\mathbf{X}]_{j\neq i}$ denote the data points in $\oplus S_{j\neq i}$. Then for any new data point $\mathbf{y} \notin \mathbf{X}$ that lies in S_i, the linear coding program

$$\min_{\bar{\mathbf{c}}_i} \|\bar{\mathbf{x}}_i - \mathbf{X}\bar{\mathbf{c}}_i\|_2^2 + \gamma\|\bar{\mathbf{c}}_i\|_\ell, \tag{15}$$

gives

$$\mathbf{y} = [\mathbf{X}]_i[\mathbf{c}^*]_i + [\mathbf{X}]_{j\neq i}[\mathbf{c}^*]_{j\neq i}, \tag{16}$$

with $[\mathbf{c}^*]_i \neq \mathbf{0}$ and $[\mathbf{c}^*]_{j\neq i} = \mathbf{0}$, where $\|\cdot\|_\ell$ denotes ℓ_1-norm or ℓ_2-norm, and $[\mathbf{c}^*]_i$ is the coefficients that associate with $[\mathbf{X}]_i$.

Proof: For any new data point $\mathbf{y} \neq \mathbf{0}$ drawn from S_i, assume there exists a non-zero vector $[\mathbf{c}^*]_{j\neq i}$ such that

$$\mathbf{y} = [\mathbf{X}]_i[\mathbf{c}^*]_i + [\mathbf{X}]_{j\neq i}[\mathbf{c}^*]_{j\neq i}, \tag{17}$$

then

$$\mathbf{y} - [\mathbf{X}]_i[\mathbf{c}^*]_i = [\mathbf{X}]_{j\neq i}[\mathbf{c}^*]_{j\neq i}. \tag{18}$$

Since $\mathbf{y} \in S_i$, then $\mathbf{y} - [\mathbf{X}]_i[\mathbf{c}^*]_i \in S_i$, i.e.,

$$[\mathbf{X}]_{j\neq i}[\mathbf{c}^*]_{j\neq i} \in S_i. \tag{19}$$

Since $S_i \cap \oplus S_{j\neq i} = 0$ and $\mathbf{y} \neq \mathbf{0}$, then, $[\mathbf{c}^*]_{j\neq i} = \mathbf{0}$ and $[\mathbf{c}^*]_i \neq \mathbf{0}$. ■

Based on Theorems 1 and 2, it is easy to obtain the following theorem.

Theorem 3: Assume that the in-sample data $\mathbf{X}$ is sufficient such that $rank(\mathbf{X}) = rank(\mathbf{Y})$. If the subspaces are linearly independent and the data are noise-free or contaminated by sample-specific corruption, then SLRR could perfectly segment the data into different subspaces.

Proof: Theorem 1 and Theorem 2 show that the affinity matrices of in-sample data and out-of-sample data are two diagonal matrices. In other words, SLRR could exactly group the data into different clusters, as desired. ■

In the similarly way, it is easy to prove the effectiveness of SSSC. Fig. 2 gives an example to show the correctness of the theorems. Specifically, we construct two independent subspaces $\{S_i\}_{i=1}^2 \in \mathbb{R}^{512}$ that consists of 512 data points. Each subspace contains 256 samples of which each one could represent the corresponding subspace. The in-sample data contains 168 data points randomly selected from the original data set. Fig. 2(c), Fig. 2(d) and Fig. 2(f) show that SSSC and SLRR disconnect the edges between inter-subspace data points. This verifies the effectiveness of the above theorems.

C. Complexity Analysis

Suppose p samples are randomly selected from n data points with dimensionality m, SSSC needs $O(t_1p^2m^2+t_1mp^3+p^2+t_2pk^2)$ to get the cluster membership of in-sample data when Homotopy optimizer [22] is used to solve ℓ_1-minimization problem and Lanczos eigensolver is used to compute the eigenvectors of Laplacian matrix $\mathbf{L}$, where k is the number of desired clusters, and t_1 and t_2 are the number of iterations of Homotopy optimizer and k-means clustering, respectively.

To group out-of-sample data points, SSSC needs computing the pseudo-inverse of the matrix $\mathbf{XX}^T$ to get the linear representation of $\bar{\mathbf{X}} \in \mathbb{R}^{m\times(n-p)}$. Therefore, the time complexity is $O(pm^2 + p^3 + (n-p)p^2)$.

Putting everything together, the computational complexity of SSSC is $O(t_1mp^3 + t_2pk^2 + np^2)$ owing to $k, m < p \ll n$. The cost of SSSC is largely less than that of SSC ($O(t_1mn^3 + t_2nk^2)$). On the other hand, the space complexity of SSSC is only $O(mp^2)$, comparing with $O(mn^2)$ of SSC.

For SLRR, the computational complexity is $O(t_3(pm^2 + p^3)+t_2p^3+np^2)$ and the space complexity is $O(mp^2)$, whereas that of LRR are $O(t_3(m^2n+n^3)+t_2nk_3^2)$ and $O(mn^2)$, where t_3 is the number of iterations of ALM.

IV. Experimental Verification and Analysis

In this section, several experiments were conducted to show the effectiveness and efficiency of Scalable Sparse Subspace Clustering (SSSC) and Scalable Low Rank Representation (SLLR).

TABLE II
DATA SETS USED IN OUR EXPERIMENTS. THE NUMBER IN THE PARENTHESES DENOTES THE RETAINING ENERGY.

Data sets	# samples	Dim.	# features	# classes
AR [45]	1400	19800	167 (98%)	100
ExYaleB [46]	2414	32256	114 (98%)	38
MPIE [47]	8916	8200	115 (98%)	286
RCV [48]	8293	18933	785 (85%)	65
PenDigits	10992	16	16 (100%)	10
Covtype [49]	581012	54	54 (100%)	7
PokerHand-1 [50]	971329	10	10 (100%)	3
PokerHand-2 [50]	1000000	10	10 (100%)	10

A. Data sets

We carried out the experiments using seven real-world data sets which cover facial images, handwritten digital data, news corpus, etc. The data sets consist of two small-sized data sets, three medium-sized data sets, and two large scale data sets. We presented some statistics of these data sets in TABLE II and a brief description as follows.

Facial images are generally believed lying onto the low-dimensional manifold, which naturally satisfy the Assumption 1. In the experiments, we used three popular face databases, i.e., AR database [45], Extended Yale database B (ExYaleB) [46], and Multi-PIE database (MPIE) [47]. AR database includes over 4000 face images of 126 people (70 male and 56 female) which vary in expression, illumination and disguise (wearing sunglasses or scarves). Each subject has 26 images consisting of 14 clean images, 6 images with sunglasses and 6 images with scarves. As did in [20], a subset, which contains 1400 clean faces randomly selected from 50 male subjects and 50 female subjects, is used in our experiment. MPIE contains the facial images of 286 individuals captured in four sessions with simultaneous variations in pose, expression and illumination. In our experiments, all images from four sessions are evaluated. Moreover, for computational efficiency, we resized AR images from 165×120 to 55×40 $(1/9)$, ExYaleB images from 192×168 to 48×42 $(1/16)$, and MPIE images from 100×82 to 50×41 $(1/4)$, and performed PCA over the cropped data to retain 98% energy of the cropped data.

Reuters-21578 (RCV) [48] is a documental corpus, we used 785 features that retain 85% information of the original data in the tests. Moreover, we carried out the experiments using three widely-used UCI data sets[2], i.e., PenDigits, Covtype [49], and PokerHand [50]. PokerHand is an extreme unbalanced data set, of which the maximal class contains 501,209 samples, comparing with 3 samples of the minimal class. We examined the performance of the algorithms using the original data set (PokerHand-1) and a subset (PokerHand-2) with 971,329 data points from the three largest subjects.

B. Baselines and Evaluation Metrics

Spectral clustering and kernel-based clustering methods are two popular methods to cope with linear inseparable data, and several studies [3] have established the equivalence between of them. In the experiments, we compared SSSC and SLRR with four scalable spectral clustering algorithms (KASP [31], Nyström approximation-based spectral clustering [28], [30], LSC [32], and SEC [35]) and one kernel-based approach (AKK [42]). Moreover, we reported the results of k-means clustering as a baseline.

We investigated the performance of two variants of Nyström-based methods and LSC, denoted as Nyström, Nyström-Orth, LSC_R, and LSC_K. The affinity matrix of Nyström is non-orthogonal, whereas that of Nyström-Orth has orthogonalized columns. LSC_R randomly selects some landmarks from data set, whereas LSC_K uses the cluster centers of k-means as in-sample data. Moreover, SEC obtains the cluster results by performing k-means in the embedding space. All algorithms were implemented in MATLAB and ran on an Intel Xeon 2.13GHz processor with 16.00 GB RAM. The used databases and the MATLAB codes of SSSC and SLRR can be downloaded at https://www.dropbox.com/sh/j5d74nzddul07u1/R4vhKvRHt9.

These tested algorithms take two approaches to speed up clustering algorithms. Specifically, SSSC, SLRR, Nyström, Nyström_Orth, LSC_R, SEC and AKK identify in-sample data by performing uniform random sampling method, whereas KASP and LSC_K use the cluster centers of k-means as in-sample data. To avoid the difference in data partitions, we pre-partitioned each database into two parts, in-sample data and out-of-sample data. After that, different algorithms were ran over the same partitions.

We measure the clustering quality by using *Accuracy* [51] and Normalized Mutual Information (*NMI*) [48] between the produced clusters and the ground truth categories. The value of *Accuracy* or *NMI* is 1 indicates perfect matching with the true subspace distribution whereas 0 indicates totally mismatch.

For a given data set $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$, let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ be the predicted labels and $\Phi = \{\phi_1, \phi_2, \ldots, \phi_n\}$ be the ground truth. The *Accuracy* is calculated via

$$\textit{Accuracy}(\Omega, \Phi) = \frac{1}{n}\sum_{i=1}^{n}\delta(\omega_i, map(\phi_i)), \tag{20}$$

where $\delta(x, y)$ equals 1 if $x = y$ and equals 0 otherwise, and $map(\phi_i)$ is the permutation mapping function that map each cluster label ϕ_i to the equivalent label from the ground truth. Generally, Kuhn-Munkres algorithm [52] is adopted to compute the best mapping.

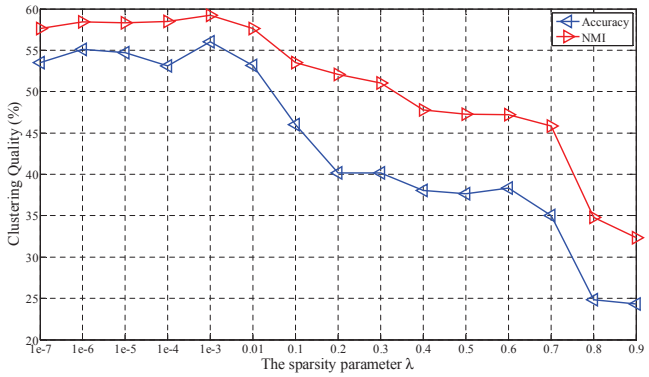
On the other hand, *NMI* is defined as

$$\textit{NMI}(\Omega, \Phi) = \frac{MI(\Omega, \Phi)}{\max(H(\Omega), H(\Phi))}, \tag{21}$$

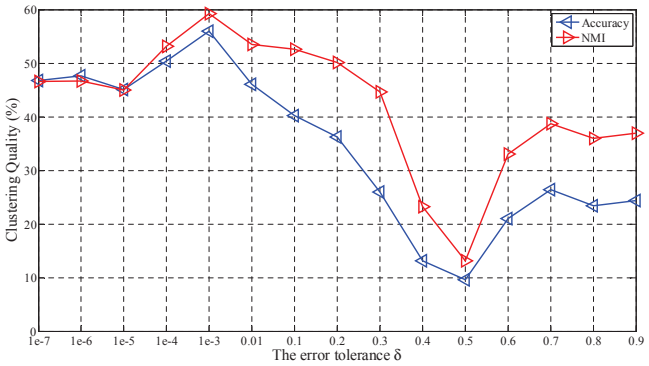
where $H(\Omega)$ and $H(\Phi)$ denote the entropies of Ω and Φ, respectively. $MI(\Omega, \Phi)$ is the mutual information between Ω and Φ, which is defined as

$$MI(\Omega, \Phi) = \sum_{\omega \in \Omega}\sum_{\phi \in \Phi} p(\omega, \phi)\log_2\left(\frac{p(\omega, \phi)}{p(\omega)p(\phi)}\right), \tag{22}$$

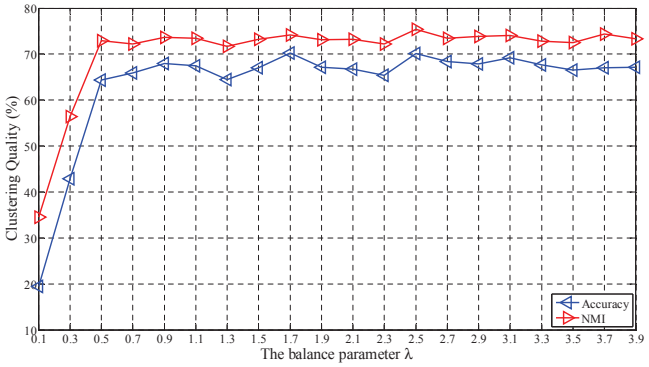
[2]http://archive.ics.uci.edu/ml/datasets.html



(a) The influences of the parameter λ of SSSC, where $\delta = e^{-3}$.



(b) The influences of the parameter δ of SSSC, where $\lambda = e^{-3}$.



(c) The influences of the parameter λ of SLRR.

Fig. 3. Clustering quality of SSSC and SLRR with varying parameters, where 1212 images are chosen from Extended Yale database B as in-sample data and the remaining are used as out-of-sample data. The x-coordinate denotes the values of the parameters, and the y-coordinate is the clustering quality (*Accuracy* and *NMI*).

where $p(\omega)$ and $p(\phi)$ are the probability distribution functions of ω and ϕ, and $p(\omega, \phi)$ denotes their joint probability distribution function.

C. Model Selection

SSSC takes $\lambda > 0$ to control the sparsity of representation and $\delta > 0$ to measure the error level of data. Moreover, SLRR uses $\lambda > 0$ to balance the effects of the two parts in (6). The choice of these parameters depends on the prior knowledge of the data distribution.

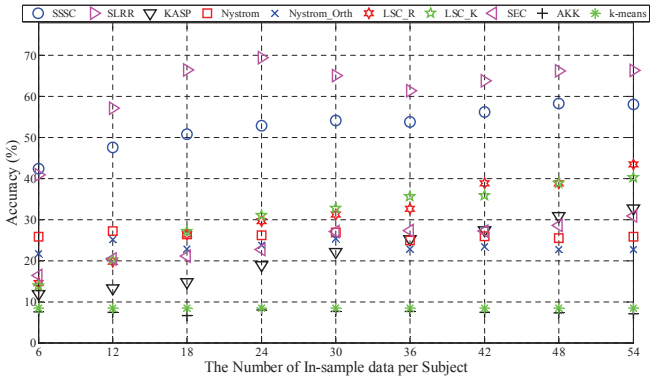
Fig. 3 shows the evaluation results of SSSC and SLRR with different values of these parameters. When λ and δ of SSSC are assigned a small positive value, e.g., from e^{-7} to 0.01, the algorithm could achieve a good results. When the parameters are assigned a big value, SSSC is failed to group the similar patterns into the same subspace. Moreover, we could see that SSSC achieved the worst results when δ equals 0.5. Although this shows the importance of model selection, the solution of SSSC can be partially stable while λ and δ is varying around a small value. On the other hand, while the λ of SLRR ranges from 0.1 to 0.3, its *Accuracy* varies from 19.47% to 42.87% and *NMI* varies from 34.48% to 56.36%. While the λ of SLRR ranges from 0.5 to 3.9, its *Accuracy* and *NMI* almost remains unchanged.

In the following experiments, we tuned the parameters of all the evaluated methods to get the highest *Accuracy*. For SSSC, we adopted Homotopy optimizer to calculate the sparse representation for each data point. The optimizer needs two user-specified parameters, sparsity parameter λ and error tolerance parameter δ. We found a good value in the ranges of $\lambda = (10^{-7}, 10^{-6}, 10^{-5})$ and $\delta = (10^{-3}, 10^{-2}, 10^{-1})$. SLRR needs pre-specifying the value of λ, and the optimal value is chosen from $[0.1, 4]$ with an interval of 0.2. Following the parameter configurations in [30], [31], [32], [42], we found the optimal parameters for the compared algorithms. In details, KASP and Nyström employ heat kernel to calculate the pairwise similarity, which need specifying the width of heat kernel τ. We specified the range of τ as $[0.1, 1]$ with an interval of 0.1 and $[2, 20]$ with an interval of 1; AKK employs RBF kernel with the parameter σ whose value range is $[0.1, 1]$ with an interval of 0.1; SEC requires three user-specified parameters, the size of neighborhood r, balance parameters μ and γ. Referring to [35], we set $\gamma = 1$ and determined the value of μ in the range of $[10^{-9}, 10^{-6}, 10^{-3}, 10^{0}, 10^{+3}, 10^{+6}, 10^{+9}, 10^{+12}, 10^{+15}]$. Moreover, KASP, LSC, SEC need to determine the size of neighborhood (r), and we found a good r in the range of $[2, 20]$ with an interval of 1.

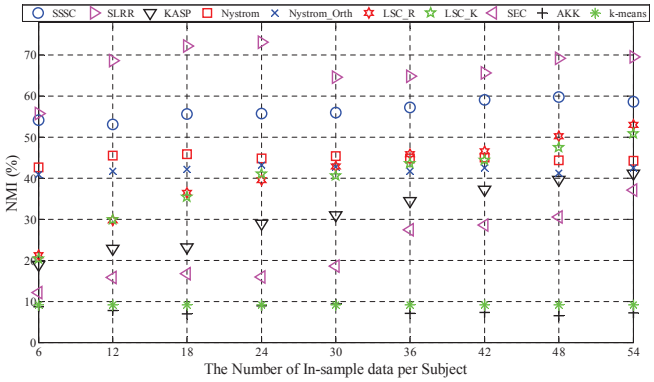
D. Experimental Results

The influence of in-sample data size: To study the influences of the in-sample data size p over clustering quality, we performed experiments over the ExYaleB database by setting $p = 38 \times \tilde{p}$, where $\tilde{p}$ denotes the sample size per subject and it increases from 6 to 54 with an interval of 6. Fig. 4 reports the clustering quality of the proposed methods, from which we have the following observations:

- All accelerating clustering methods except SEC and AKK outperform the baseline algorithm (k-means) in *Accuracy* and *NMI*. SSSC and SLRR are superior to the other investigated approaches by a considerable performance margin. Specifically, SSSC achieves 14.58% gain in *Accuracy* and 5.70% gain in *NMI* over the second best algorithm (LRC_R) when 2052 (54 samples per class) images are selected as in-sample data. SLRR achieves 15.08% gain in *Accuracy* and 13.12% gain in *NMI* over the second best algorithm (Nyström) when 228 (6 samples per class) images are selected as in-sample data.



(a) *Accuracy* versus the varying number of in-sample data



(b) *NMI* versus the varying number of in-sample data

Fig. 4. Clustering quality of the competing algorithms using the Extended Yale database B. The x-coordinate denotes the number of in-sample data per subject, and the y-coordinate is the clustering quality (*Accuracy* or *NMI*).

- All these scalable algorithms perform better with increasing $\tilde{p}$, except Nyström and Nyström_Orth which cope with the scalability issues of spectral clustering by reducing the size of affinity matrix but that of data set.

Small-sized data sets: To examine the effectiveness of our methods in handling new samples, we carried out the experiments using the AR and ExYaleB databases. Moreover, we reported the results of k-means, SSC, and LRR over the whole data sets. From Tables III-IV, we could find that

- Our framework successfully makes SSC and LRR feasible to grouping new data with acceptable loss in clustering quality. For example, the *Accuracy* of SSC is 8.57% higher than that of SSSC, whereas the time cost of SSC is about four times that of SSSC. With the increase of data size, SSC and LRR will be failed to get the results, whereas SSSC and SLRR could group these new coming data without recalculating the affinity matrix and cluster membership of the whole data set.
- Comparing with the other scalable methods (KASP, Nyström, Nyström_Orth, LSC_R, LSC_K, SEC, and AKK), SSSC and SLRR find an elegant balance between clustering quality and time costs. Although SSSC and SLRR are not the fastest algorithm, they achieve the best results.
- SLRR performs better than SSSC in these tests. The possible reason is that the low rank representation could capture the relationship among data points, whereas sparse representation cannot, as suggested in [17].
- In [35], Nie et al. reported the results of SEC using the ExYaleB data set. The highest *Accuracy* of SEC is about 42.8% in their tests, comparing with 24.65% in our experiment. The potential reason for the performance difference is that they adopted spectral rotation to get the cluster membership, whereas we adopted k-means for all competing methods. Note that, the best result (42.8%) of SEC reported in their work is still lower than the results achieved by SSSC (55.88%) and SLRR (67.19%).

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING **AR**, WHERE A HALF OF IMAGES (700) ARE SELECTED AS IN-SAMPLE DATA. THE NUMBERS IN THE PARENTHESIS ARE THE TUNED PARAMETERS.

Algorithms	Accuracy	NMI	Time (s)
SSSC	63.07% (1e-5, 1e-3)	82.61%	142.22
SLRR	**72.79%** (3.1)	**88.08%**	40.12
KASP [31]	34.50% (0.1)	63.73%	134.81
Nyström [30]	61.29% (2)	83.33%	2.34
Nyström_Orth [30]	59.43% (0.9)	79.84%	13.71
LSC_R [32]	32.07% (4)	62.31%	1.67
LSC_K [32]	35.29% (4)	64.16%	2.20
SEC [35]	29.29% (1e+9, 8)	45.33%	1.72
AKK [42]	25.14% (0.2)	54.29%	0.76
SSC [12], [13]	71.64% (1e-7, 1e-3)	86.47%	461.41
LRR [15]	78.43% (1)	89.86%	152.93
k-means [2]	28.64%	59.05%	18.79

Note: SSC, LRR, and k-means cannot handle out-of-sample data. Thus, the results of SSC, LRR, and k-means are achieved by directly performing them over the whole data set.

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING **EXYALEB**, WHERE A HALF OF IMAGES (1212) ARE SELECTED AS IN-SAMPLE DATA. THE NUMBERS IN THE PARENTHESIS ARE THE TUNED PARAMETERS.

Algorithms	Accuracy	NMI	Time (s)
SSSC	55.88% (1e-3, 1e-3)	57.91%	128.05
SLRR	**67.19%** (2.9)	**73.16%**	26.83
KASP [31]	22.54% (8)	32.50%	37.77
Nyström [30]	25.27% (12)	45.19%	8.22
Nyström_Orth [30]	20.96% (3)	40.21%	60.89
LSC_R [32]	31.48% (2)	54.98%	7.26
LSC_K [32]	33.76% (2)	54.30%	8.25
SEC [35]	24.65% (1e-9, 1)	39.55%	10.35
AKK [42]	7.25% (0.4)	6.87%	3.02
SSC [12], [13]	62.64% (1e-5,0.01)	65.30%	344.93
LRR [15]	71.17% (2)	75.67%	226.83
k-means [2]	8.82%	10.24%	50.23

Medium-sized data sets: To examine the performance the SSSC and SLRR over different types of data, we carried out the experiments using MPIE (facial images), RCV (text data), and PenDigits (handwritten digital data). It should be pointed out that we didn't report the performance of SSC and LRR because they have been failed to achieve results over these databases within an acceptable time cost. Tables V-VII report the clustering quality and the time cost of these methods, from which we can conclude

- SSSC and SLRR generally outperform the other scalable methods by a considerable performance margin. In details, SSSC achieves a 9.61% gain in *Accuracy* on MPIE

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING **MPIE**, WHERE 1000 IMAGES ARE SELECTED AS IN-SAMPLE DATA. THE NUMBERS IN THE PARENTHESIS ARE THE TUNED PARAMETERS.

Algorithms	Accuracy	NMI	Time (s)
SSSC	61.53% (1e-6, 0.01)	82.81%	847.06
SLRR	**66.56%** (2.3)	**85.09%**	340.44
KASP [31]	17.09% (0.1)	57.54%	1479.83
Nyström [30]	47.01% (0.7)	76.97%	15.28
Nyström_Orth [30]	51.92% (0.7)	79.96%	64.80
LSC_R [32]	18.50% (2)	54.90%	62.05
LSC_K [32]	17.50% (3)	56.61%	65.69
SEC [35]	17.95% (1e-3, 9)	43.73%	27.20
AKK [42]	11.40% (0.1)	40.16%	24.62
k-means [2]	14.69%	52.45%	268.54

TABLE VI
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING **RCV**, WHERE 1500 IMAGES ARE SELECTED AS IN-SAMPLE DATA. THE NUMBERS IN THE PARENTHESIS ARE THE TUNED PARAMETERS.

Algorithms	Accuracy	NMI	Time (s)
SSSC	32.40% (1e-7,0.01)	33.81%	1320.63
SLRR	**39.28%** (3.10)	13.24%	499.57
KASP [31]	22.32% (0.1)	24.79%	198.80
Nyström [30]	23.22% (0.4)	27.55%	27.08
Nyström_Orth [30]	25.88% (0.1)	22.70%	3401.30
LSC_R [32]	14.24% (2)	22.58%	8.87
LSC_K [32]	14.45% (2)	23.69%	17.72
SEC [35]	17.56% (1e-6,3)	26.14%	19.87
AKK [42]	23.57% (0.2)	**36.40%**	27.94
k-means [2]	19.05%	26.98%	0.33

TABLE VII
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING **PENDIGITS**, WHERE 1000 IMAGES ARE SELECTED AS IN-SAMPLE DATA. THE NUMBERS IN THE PARENTHESIS ARE THE TUNED PARAMETERS.

Algorithms	Accuracy	NMI	Time (s)
SSSC	**81.99%** (1e-7, 1e-4)	**78.37%**	17.02
SLRR	75.38% (0.3)	68.86%	10.42
KASP [31]	77.84% (4)	77.97%	12.48
Nyström [30]	77.96% (0.4)	70.20%	35.93
Nyström_Orth [30]	74.78% (3)	67.59%	6.20
LSC_R [32]	80.09% (15)	76.67%	5.62
LSC_K [32]	81.73% (11)	77.34%	7.93
SEC [35]	78.09% (1e-9, 4)	71.12%	11.83
AKK [42]	77.02% (0.01)	69.15%	6.21
k-means [2]	77.05%	69.21%	23.70

TABLE VIII
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS USING **COVTYPE**.

Algorithms	Accuracy	NMI	Time (s)
SSSC	**31.05%** (1e-5, 0.1)	**6.82%**	325.51
SLRR	25.58% (0.1)	4.61%	240.91
KASP [31]	25.36% (3)	3.50%	1314.52
Nyström [30]	23.76% (0.1)	3.79%	40.61
Nyström_Orth [30]	23.36% (0.1)	3.98%	351.58
LSC_R [32]	23.24% (2)	6.06%	154.48
LSC_K [32]	25.90% (4)	6.74%	1155.40
SEC [35]	21.65% (1, 4)	3.87%	64.85
AKK [42]	25.36% (1)	3.67%	344.24
k-means [2]	20.84%	3.69%	4895.70

over the second best algorithm (Nyström), and the gain achieved by SLRR is about 14.64%.

- The running times is a main weakness of SSSC and SLRR even though they are more efficient than the original algorithms (SSC and LRR). In the experiments, we found that most time was used to group in-sample data. For example, 1320.63 seconds were taken to group the whole RCV data set, of which 1100 seconds are used to perform SSC over in-sample data. As grouping of in-sample data is an offline process, we assume that our algorithms are more competitive in large scale setting as demonstrated in the next experiment. Note that, SSC and LRR have been failed to achieve the results over these three data sets.
- In most cases, LSC_K outperforms LSC_R with a little improvement, which verifies the claim [53] that the complex sampling techniques actually could not produce a better result than random sampling method.
- In [32], Chen and Cai investigated the *Accuracy* of LSC_R, LSC_K, Nyström_Orth, and KASP using the PenDigits database. The highest *Accuracy* of these algorithms are 79.04%, 79.27%, 73.94% and 72.47%, comparing with 81.73%, 80.09%, 74.78% and 77.84% achieved in our experiments.

Large scale data sets: Tables VIII-X show the results of the investigated approaches over three large scale data sets. For each data set, 1000 samples are selected as in-sample data, and the remaining samples are used as out-of-sample data. We have the following observations:

- SSSC and SLRR are superior to the other approaches. For example, the *Accuracy* of SSSC over Covtype is at least 5.14% higher than the other tested methods. For PokerHand-1 (only contains the three largest subjects of PokerHand-2) and PokerHand-2 (the whole data set), the gains are 2.83% and 3.71%, respectively.
- The metric *NMI* cannot distinct the performance of the evaluated algorithms over these databases, where the score of *NMI* achieved by the tested methods are close to 0.
- In [32], the highest *Accuracy* over Covtype achieved by LSC_R, LSC_K, Nyström_Orth and KASP is 24.75%, 25.50%, 22.31% and 22.42%, respectively. In our experiments, the *Accuracy* of these four algorithms are 21.76%, 21.79%, 24.66% and 24.18%, respectively. The possible reason may be attributed to the difference in the choice of in-sample data.
- With the increase of data size, SSSC and SLRR demonstrates a good balance between running time and clustering quality. Moreover, the used memory of our methods only depends on the number of in-sample data, which makes SSSC and SLRR are competitive in large scale setting.

V. CONCLUSION

The representation-based spectral clustering algorithms have became more and more popular owing to its effectiveness. However, the over-high computational complexity and offline computing requirement have hindered their application in practice. Especially, the scenario of booming big data. In this paper,

TABLE IX
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **POKERHAND-1**.

Algorithms	Accuracy	NMI	Time (s)
SSSC	**51.60%** (1e-7, 0.2)	**0.63%**	267.71
SLRR	39.85% (0.10)	0.14%	166.89
KASP [31]	35.24% (0.3)	0.09%	5497.06
Nyström [30]	47.90% (0.2)	0.16%	61.43
Nyström_Orth [30]	47.74% (20)	0.00%	204.43
LSC_R [32]	34.91% (8)	0.00%	1891.00
LSC_K [32]	34.99% (2)	0.00%	8765.50
SEC [35]	33.34% (1e-9,3)	0.00%	81.38
AKK [42]	35.96% (0.1)	0.62%	1039.28
k-means [2]	36.02%	0.01%	4760.40

TABLE X
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **POKERHAND-2**.

Algorithms	Accuracy	NMI	Time (s)
SSSC	**19.31%** (1e-5, 0.1)	**0.20%**	474.14
SLRR	16.64% (0.1)	0.09%	317.69
KASP [31]	12.37% (3)	0.05%	7049.90
Nyström [30]	13.09% (0.2)	0.23%	65.50
Nyström_Orth [30]	16.48% (17)	0.04%	205.70
LSC_R [32]	12.24% (5)	0.00%	1936.80
LSC_K [32]	12.32% (3)	0.00%	8829.00
SEC [35]	14.51% (1e-3, 4)	0.08%	130.24
AKK [42]	10.50% (0.01)	0.03%	2882.50
k-means [2]	10.41%	0.03%	7188.80

we have presented a general framework to address the stability issue and out-of-sample problem in these algorithms and successfully applied our approach to SSC and LRR. The proposed algorithms, Scalable Sparse Subspace Clustering (SSSC) and Scalable Low Rank Representation (SLRR), which resolve the scalability issue and out-of-sample problem of SSC and LRR simultaneously. SSSC (SLRR) reduces the time complexity of SSC (LRR) from cube to linearity of problem size, while preserves good performance. Extensive experiments show the effectiveness and efficiency of our approach, comparing with the state-of-the-art methods.

There is no work is perfect, and this work is no exception. Although we have proved that SSSC and SLRR could achieve perfect data segmentations when subspaces are independent and only a small fraction of samples are corrupted, it is also interesting and challenging to explore the recovery conditions or develop the error bound for SSSC and SLRR in more complex situations, e.g., the subspaces are dependent or the data are grossly-corrupted.

ACKNOWLEDGMENT

REFERENCES

[1] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip *et al.*, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.

[2] D. Cai, "Litekmeans: the fastest matlab implementation of kmeans," *Available at: http://www.zjucadcg.cn/dengcai/Data/Clustering.html*, 2011.

[3] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 176–190, 2008.

[4] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems*, vol. 14, no. 1, pp. 849–856, 2002.

[5] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 1, p. 1, 2009.

[6] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[7] Y.-H. Chu, J.-W. Huang, K.-T. Chuang, D.-N. Yang, and M.-S. Chen, "Density conscious subspace clustering for high-dimensional data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 1, pp. 16–30, jan. 2010.

[8] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001.

[9] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1546–1562, 2007.

[10] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1-12, pp. 743–750, 2008.

[11] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[12] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.

[13] ——, "Sparse subspace clustering: Algorithm, theory, and applications," *To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence, arXiv:1203.1005*, 2012.

[14] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang, "Learning with ℓ^1-graph for image analysis," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 858–866, 2010.

[15] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[16] G. C. Liu and S. C. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," *2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 1615–1622, 2011.

[17] R. Liu, Z. Lin, F. D. la Torre, and Z. Su, "Fixed-rank representation for unsupervised visual learning," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 598–605.

[18] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *European Conference on Computer Vision*, vol. 7578, 2012, pp. 347–360.

[19] X. Peng, L. Zhang, and Z. Yi, "Constructing l2-graph for subspace learning and segmentation," *CoRR*, vol. abs/1209.0841, 2012.

[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[21] A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Fast l1-minimization algorithms and an application in robust face recognition: A review," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-13, February 5 2010.

[22] M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *Ima Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389–403, 2000.

[23] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[24] E. Cands and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[25] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *arXiv preprint arXiv:0912.3599*, 2009.

[26] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[27] L. Wang, M. Rege, M. Dong, and Y. Ding, "Low-rank kernel matrix factorization for large-scale evolutionary clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 1036–1050, June 2012.

[28] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[29] M. Li, J. T. Kwok, and B.-L. Lu, "Making large-scale nyström approximation possible." in *International Conference on Machine Learning*, J. Frnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 631–638.

[30] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 568–586, 2011.

[31] D. H. Yan, L. Huang, and M. I. Jordan, "Fast approximate spectral clustering," *Kdd-09: 15th Acm Sigkdd Conference on Knowledge Discovery and Data Mining*, pp. 907–915, 2009.

[32] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *AAAI Conference on Artificial Intelligence*, 2011.

[33] L. Wang, C. Leckie, R. Kotagiri, and J. Bezdek, "Approximate pairwise clustering for large data sets via sampling plus extension," *Pattern Recognition*, vol. 44, no. 2, pp. 222–235, 2011.

[34] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[35] F. Nie, Z. Zeng, T. I. W., D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1796–1808, 2011.

[36] Z. Y. Xi Peng, Lei Zhang, "Scalable sparse subspace clustering," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2013.

[37] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems*, 2011.

[38] G. Liu and S. Yan, "Active subspace: Toward scalable low-rank learning," *Neural Computation*, pp. 1–24, 2012.

[39] X. Zhang, F. Sun, G. Liu, and Y. Ma, "Fast low-rank subspace segmentation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, no. PrePrints, p. 1, 2013.

[40] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *IEEE International Conference on Computer Vision*, vol. 2. IEEE, 2005, pp. 1208–1213.

[41] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.

[42] R. Chitta, R. Jin, T. Havens, and A. Jain, "Approximate kernel k-means: solution to large scale kernel clustering," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 895–903.

[43] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *IEEE International Conference on Computer Vision*, 2011.

[44] X. Peng, L. Zhang, Z. Yi, and K. K. Tan, "Learning locality-constrained collaborative representation for face recognition," *CoRR*, vol. abs/1210.1316, 2012.

[45] A. Martinez, "The ar face database," *CVC Technical Report*, vol. 24, 1998.

[46] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[47] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

[48] D. Cai, X. F. He, and J. W. Han, "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.

[49] F. Alimoglu and E. Alpaydin, "Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition," in *Turkish Artificial Intelligence and Artificial Neural Networks Symposium*, vol. 1, 1996.

[50] J. Blackard and D. Dean, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables," *Computers and Electronics in Agriculture*, vol. 24, no. 3, pp. 131–151, 1999.

[51] Y. Zhao and G. Karypis, "Criterion functions for document clustering: Experiments and analysis," *Machine Learning*, 2001.

[52] L. Lovsz and M. Plummer, *Matching theory*. American Mathematical Society, 2009, vol. 367.

[53] T. O. Kvalseth, "Entropy and correlation: Some comments," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 17, no. 3, pp. 517–519, 1987.