

A Unified Framework for Representation-Based Subspace Clustering of Out-of-Sample and Large-Scale Data

Xi Peng, Huajin Tang, *Member IEEE*, Lei Zhang, *Member IEEE*, Zhang Yi, *Senior Member IEEE*, Shijie Xiao *Student Member IEEE*,

Abstract—Under the framework of spectral clustering, the key of subspace clustering is building a similarity graph which describes the neighborhood relations among data points. Some recent works build the graph using sparse, low-rank, and ℓ_2 -norm-based representation, and have achieved state-of-the-art performance. However, these methods have suffered from the following two limitations. First, the time complexities of these methods are at least proportional to the cube of the data size, which make those methods inefficient for solving large-scale problems. Second, they cannot cope with out-of-sample data that are not used to construct the similarity graph. To cluster each out-of-sample datum, the methods have to recalculate the similarity graph and the cluster membership of the whole data set. In this paper, we propose a unified framework which makes representation-based subspace clustering algorithms feasible to cluster both out-of-sample and large-scale data. Under our framework, the large-scale problem is tackled by converting it as out-of-sample problem in the manner of “sampling, clustering, coding, and classifying”. Furthermore, we give an estimation for the error bounds by treating each subspace as a point in a hyperspace. Extensive experimental results on various benchmark data sets show that our methods outperform several recently-proposed scalable methods in clustering large-scale data set.

Index Terms—Scalable subspace clustering, out-of-sample problem, sparse subspace clustering, low-rank representation, least square regression, error bound analysis.

I. INTRODUCTION

CLUSTERING analysis aims to group 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 have been proposed, for example, partitioning-based clustering [1], kernel-based clustering [2], and subspace clustering [3].

Subspace clustering aims at finding a low-dimensional subspace to fit each group of data points. It mainly contains two tasks, *i.e.*, projecting the data set into another space (encoding) and calculating the cluster membership of the data set in the projection space (clustering). Popular subspace clustering methods include but not limit to statistical methods [4], [5] and spectral clustering [6], [7]. Spectral clustering finds the cluster

membership of the data points by using the spectrum of an affinity matrix. The affinity matrix corresponds to a similarity graph of which each vertex denotes a data point, with the edge weights representing the similarities between connected points. Thus, at the heart of the spectral clustering is a similarity graph construction problem.

There are two widely-used approaches to build a similarity graph, *i.e.*, Pairwise Distance (PD) and Reconstruction Coefficients (RC). Specifically, PD computes the similarity based on the distance (*e.g.*, the Euclidean distance) between any two data points. However, PD cannot reflect the global structure of the data set, because its value only depends on connected data points. In contrast, RC denotes each data point as a linear combination of the other points and uses the representation coefficients as a similarity measurement. Several recent works have shown that RC is superior to PD in subspace clustering, for example, sparse representation [8], [9], [10], [11], [12], [13], low rank representation [14], [15], [16], [17], latent low rank representation [18], and ℓ_2 -norm-based representation [19], [20].

Although representation-based subspace clustering has been extensively studied, how to solve the large-scale and out-of-sample clustering problems are less explored. Taking sparse subspace clustering (SSC) [8], [9] as an example: SSC iteratively computes the sparse codes of n data points and performs eigen-decomposition over an $n \times n$ graph Laplacian matrix. Its computational complexity is more than $O(mn^3)$ even though the fastest ℓ_1 -solver is used, where m denotes the dimensionality of the data set. Thus, any medium-sized data set will bring up large-scale problem with SSC. Moreover, SSC cannot handle out-of-sample data that are not used to construct the similarity graph. To cluster each previously unseen datum¹, SSC has to recompute the similarity graph and the cluster membership of the whole data set. In fact, most representation-based subspace clustering methods [14], [15], [17], [18], [19], [21] have suffered from similar limitations when dealing with large-scale or out-of-sample data.

To address such issues, we propose a unified framework for the representation-based subspace clustering algorithms. Our framework treats the large-scale problem as the out-of-sample problem in the manner of “sampling, clustering, coding, and classifying” (Fig. 1). Specifically, we split a large

X. Peng is with Institute for Infocomm Research, A*STAR, Singapore 138632 (E-mail: pangsaai@gmail.com).

H. Tang, L. Zhang and Z. Yi are with the College of Computer Science, Sichuan University, Chengdu, China 610065 (E-mail: htang@i2r.a-star.edu.sg, leizhang@scu.edu.cn, zhangyi@scu.edu.cn).

S. J. Xiao is with School of Computer Engineering, Nanyang Technological University, Singapore (E-mail:xiao0050@ntu.edu.sg).

Corresponding author: H. Tang (E-mail: htang@i2r.a-star.edu.sg).

¹In this paper, we assume that any previously unseen datum (*i.e.*, out-of-sample datum) belongs to one of the subspaces spanned by in-sample data.

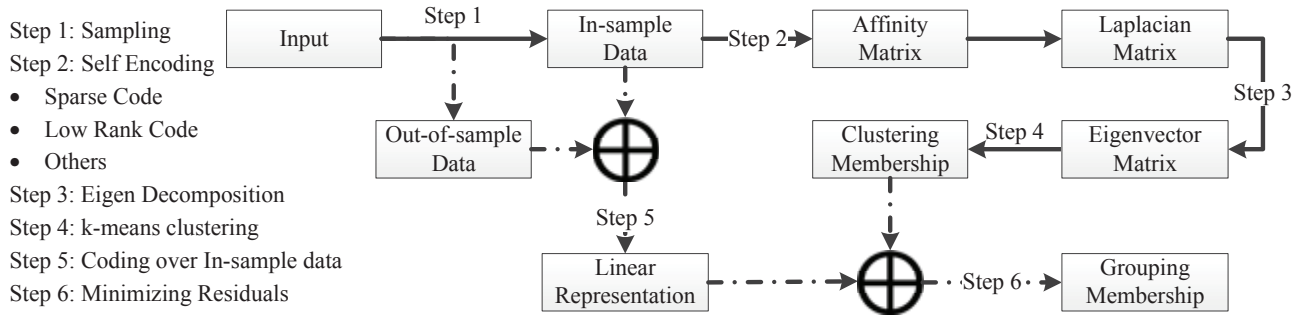


Fig. 1. Architecture of the proposed framework for scalable representation-based subspace clustering. The framework can be summarized as “sampling (step 1), clustering (steps 2–4), coding (step 5), and classifying (step 6)”. Solid and dotted lines are used to show the processes of clustering of in-sample data and out-of-sample data, respectively. For the out-of-sample problem, only steps 5 and 6 are needed.

scale data set into two parts, in-sample data \mathbf{X} and out-of-sample data \mathbf{Y} . Then, we obtain the cluster membership of \mathbf{X} and assign each out-of-sample datum to the nearest subspace spanned by \mathbf{X} . Under our framework, three scalable methods are presented, *i.e.*, scalable sparse subspace clustering (SSSC), scalable low rank representation (SLRR), and scalable least square regression (SLSR). The proposed methods remarkably improve the computational efficiency of the original approaches while preserving a good clustering performance.

This paper is a substantial extension of our conference paper [22], which is further improved from the following aspects: 1) We perform error analysis for our framework by treating each subspace in a well-defined hyperspace. The presented lower and upper error bounds are helpful in understanding the working mechanism of the nearest subspace classifier (specifically, sparse representation based classifier (SRC) [23]. To the best of our knowledge, this is the first work to perform errors analysis for SRC. 2) We additionally propose two scalable methods, *i.e.*, SLRR and SLSR, which make low rank representation (LRR) [15] and least square regression (LSR) [19] feasible to cluster large scale data and out-of-sample data. 3) We perform extensive experiments to compare our methods with more scalable clustering methods on more data sets. 4) We conduct comprehensive analysis for our approaches, including the performance with different out-of-sample grouping strategies and the influence of different parameters.

The rest of the paper is organized as follows: We provide in Section II a brief review of the representation-based clustering algorithms and some scalable spectral clustering methods. In Section III, we propose our framework and three scalable representation-based clustering algorithms, and further present some theoretical results on the error bound analysis of our framework. To demonstrate the performance of our proposed methods, we compare them with five recently-proposed scalable clustering approaches on nine data sets in Section IV. Lastly, we give the conclusions and the further work in Section V.

II. REPRESENTATION-BASED SUBSPACE CLUSTERING

In this paper, we use **lower-case bold letters** to represent column vectors and **UPPER-CASE BOLD LETTERS** to

TABLE I
NOTATIONS.

Notation	Definition
n	the number of data points
m	the dimensionality of a given data set
k	the number of clusters
p	the number of in-sample data
t	the number of iterations of algorithm
r	the rank of a given data matrix
$f(\mathbf{x}_i)$	the prediction for a given \mathbf{x}_i
$\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n]$	data set
$[\mathbf{D}]_i$	the data points belonging to the subspace S_i
$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p]$	in-sample data
$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n-p}]$	out-of-sample data
$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots]$	the representation of a given data set
$\mathbf{A} \in \mathbb{R}^{n \times n}$	affinity matrix based on \mathbf{C}
$\mathbf{L} \in \mathbb{R}^{n \times n}$	Laplacian matrix
$\mathbf{V} \in \mathbb{R}^{n \times k}$	eigenvector matrix

represent matrices. \mathbf{A}^T and \mathbf{A}^{-1} denote the transpose and pseudo-inverse of the matrix \mathbf{A} , respectively. \mathbf{I} denotes the identity matrix. Table I summarizes some notations used throughout the paper.

A. Sparse Representation Based Subspace Clustering

Recently, Elhamifar and Vidal [8], [9] proposed SSC with well-founded recovery theory for independent subspaces and disjoint subspaces. SSC calculates the similarity among data points by solving the following optimization problem:

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

where $\mathbf{C} \in \mathbb{R}^{n \times n}$ is the sparse representation of the data set $\mathbf{D} \in \mathbb{R}^{m \times n}$, \mathbf{E} corresponds to the sparse outlying entries, \mathbf{Z} denotes the reconstruction errors for the limited representational capability, and the parameters λ_E and λ_Z balance these three terms in the objective function. (1) is convex and can be solved by a number of ℓ_1 -solvers [24]. After getting \mathbf{C} , SSC builds a similarity graph via $\mathbf{A} = |\mathbf{C}|^T + |\mathbf{C}|$ and performs spectral clustering [6] over the graph.

SSC is effective but inefficient. It needs $O(tn^2m^2 + tmn^3)$ to build the similarity graph even if the fastest ℓ_1 -solver is used, where t denotes the number of iterations of the solver.

Algorithm 1 Representation based Subspace Clustering.

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

- 1: Obtain the representation \mathbf{C}^* by solving (1), (2) or (3).
- 2: Get the affinity matrix via $\mathbf{A} = |\mathbf{C}^*|^T + |\mathbf{C}^*|$.
- 3: Construct a Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{B}^{-1/2} \mathbf{A} \mathbf{B}^{-1/2}$ using \mathbf{A} , where $\mathbf{B} = \text{diag}\{b_i\}$ with $b_i = \sum_{j=1}^n \mathbf{A}_{ij}$.
- 4: Obtain the eigenvector matrix $\mathbf{U} \in \mathbb{R}^{n \times k}$ 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 algorithm over the rows of \mathbf{U} .

Output: The cluster assignment of \mathbf{D} .

In addition, SSC takes $O(n^3)$ to calculate the eigenvectors of the Laplacian matrix \mathbf{L} . Considering that \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 Based Subspace Clustering

Different from SSC, LRR [15], [18], [25] uses the lowest rank representation rather than the sparsest representation to build the similarity graph. The objective function of LRR is

$$\min_{\mathbf{C}, \mathbf{E}} \|\mathbf{C}\|_* + \lambda \|\mathbf{E}\|_\ell \quad \text{s.t. } \mathbf{D} = \mathbf{D}\mathbf{C} + \mathbf{E}, \quad (2)$$

where $\|\cdot\|_*$ denotes the nuclear norm, $\|\cdot\|_\ell$ could be chosen as $\ell_{2,1}$ -norm, ℓ_1 -norm, or Frobenius norm, depending on prior knowledge of the error structure. Generally, $\ell_{2,1}$ -norm is adopted to deal with sample-specific corruption and outlier, ℓ_1 -norm is used to characterize the random corruption, and Frobenius norm is used to handle the Gaussian noise.

LRR, which adopts augmented Lagrange multipliers (ALM) method to solve (2), takes $O(m^2n + n^3)$ to perform singular value decomposition (SVD) over a dense matrix at each iteration. In addition, LRR will take $O(n^3 + t_2nk^2)$ to perform clustering, where t_2 denotes the number of iterations of the k-means method. Therefore, the overall time complexity of LRR is $O(t_1m^2n + t_1n^3 + t_2nk^2)$, where t_1 is the number of iterations of ALM.

C. ℓ_2 -norm Based Methods

SSC, LRR, and their extensions solve a convex optimization problem, of which the computational complexities are very high. Recently, least square regression (LSR) [19] has shown that ℓ_2 -norm-based representation can achieve the competitive result with faster speed. LSR aims at solving

$$\min_{\mathbf{C}} \|\mathbf{D} - \mathbf{D}\mathbf{C}\|_F^2 + \lambda \|\mathbf{C}\|_F^2 \quad \text{s.t. } \text{diag}(\mathbf{C}) = \mathbf{0}, \quad (3)$$

where $\|\cdot\|_F$ denotes the Frobenius norm, the non-negative real number λ is used to avoid overfitting, and the constraint guarantees that the i -th coefficient over $\mathbf{d}_i \in \mathbf{D}$ is zero.

Lu et al. [19] provides two solutions to (3) and the computational complexities of these solutions are $O(m^2n)$ at least.

Thus, the overall computational complexity of LSR is about $O(m^2n + n^3 + tnk^2)$, where t denotes the number of iterations of the k-means method. Clearly, LSR has also suffered from the large-scale problem as SSC and LRR did.

Besides the large scale clustering problem, SSC, LRR, and LSR have suffered from the out-of-sample problem, *i.e.*, they cannot cope with the data that are not used to construct the similarity graph. For each previously unseen datum, SSC, LRR, and LSR have to perform the algorithm over the whole data set once again. This makes them impossible to cluster incremental data. SSC, LRR, and LSR are summarized in Algorithm 1.

There are some methods have been proposed to reduce the time cost of minimizing lowest-rank matrix [26], [27]. However, the methods mainly focused on speeding up the encoding process without consideration of the clustering process.

D. Scalable Spectral Clustering Algorithms

Recently, some works have focused on solving the large-scale clustering problem of the traditional spectral clustering. One natural way is to reduce the time cost of eigen-decomposition over the Laplacian matrix. For example, [28] adopted Nyström method to get the approximation of the eigenvectors of the whole similarity matrix. [29] solved the generalized eigenvalue problem in a distributed computing platform.

Another way is reducing the data size by replacing the original data with a small number of samples. [30] presented a fast spectral clustering algorithm by selecting some representative points from the input and got the cluster assignment based on the chosen samples. [31] proposed landmark-based spectral clustering algorithm. The algorithm chooses p representative points as the landmarks and constructs a Laplacian matrix via $\mathbf{L} = \mathbf{A}^T \mathbf{A}$, where the element of $\mathbf{A} \in \mathbb{R}^{p \times n}$ is the pairwise distance between the input data and the landmarks. [32] selects the landmarks by performing selective sampling technique and running spectral clustering over the chosen samples based on pairwise distance. [33] proposed spectral embedded clustering (SEC) which groups out-of-sample data in a linear projection space. The main difference among the above works is the method to handle out-of-sample data. Different from the above sampling-based method, Belabbas and Wolfe [34] proposed a quantization based method with theoretical justification to select in-sample data in a deterministic way. By extending the quantization based method with self-organizing maps (SOMs), Tasdemir [35] recently proposed a novel method by utilizing the quantization property of SOMs and neural gas to handle the large scale data set. Extensive experimental studies show that this method has achieved impressive performance compared with sampling-based methods on a range of data sets. Although numerous works have been conducted on speeding up the pairwise distance based clustering methods, very few researches have been done to enhance the scalability of the representation based approaches.

III. SCALABLE SUBSPACE CLUSTERING AND ERROR ANALYSIS

In this section, we present our framework which makes the representation based subspace clustering methods feasible to handle large scale data and out-of-sample data. Our method treats the large-scale problem as the out-of-sample problem by taking the strategy 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 group of out-of-sample data and assign the data to the subspace that has the minimal residual. Note that, to solve the out-of-sample problem, only the last two steps are needed.

A. The Proposed Methods

Our framework is based on a general assumption as follows:

Assumption 1. *Suppose the data set $[\mathbf{D}]_i \in \mathbb{R}^{m \times n_i}$ is drawn from the subspace S_i , one could use a small portion of $[\mathbf{D}]_i$, denoted by $[\mathbf{X}]_i \in \mathbb{R}^{m \times p_i}$, to learn the structure of S_i , where $\text{rank}([\mathbf{D}]_i) = \text{rank}([\mathbf{X}]_i)$, $\text{rank}([\mathbf{X}]_i) \leq p_i \ll n_i$, and S_i is a compact metric space.*

Assumption 1 is twofold. First, it implies that each data point could be encoded as a linear combination of a few basis (*i.e.*, sparsity assumption). Second, it requires that $[\mathbf{X}]_i$ and $[\mathbf{D}]_i$ are independent and identically distributed (*i.e.*, i.i.d.) so that out-of-sample data could be represented by $[\mathbf{X}]_i$. The assumption is very general on which most data mining and machine learning works are based.

In practice, the sparsity assumption is easily satisfied for high-dimensional data such as facial images. To satisfy the assumption of i.i.d., we need to find the representative points $\mathbf{X} \in \mathbb{R}^{m \times p}$ from $\mathbf{D} \in \mathbb{R}^{m \times n}$ so that out-of-sample data $\mathbf{Y} \in \mathbb{R}^{m \times (n-p)}$ locate in the subspaces spanned by \mathbf{X} . To this end, some sampling techniques such as column selection method [36] can be used. However, these sampling methods are inefficient and cannot be applied to large scale setting. In this paper, we adopt uniform random sampling approach of which time cost is only $O(1)$. In addition to computational efficiency, the uniform random sampling method can perform comparably to the complex sampling techniques as shown in [30], [33]. After sampling and getting the cluster membership of in-sample data \mathbf{X} , we handle out-of-sample data \mathbf{Y} based on the knowledge learnt from \mathbf{X} . The simplest approach is assigning each $\mathbf{y}_i \in \mathbf{Y}$ to the nearest $\mathbf{x}_j \in \mathbf{X}$ in terms of the Euclidean distance. However, such approach implicitly requires some prior knowledge. For example, the data set must locate in the Euclidean space otherwise \mathbf{y}_i would not be correctly clustered.

In this work, we compute the sparse representation of \mathbf{Y} over \mathbf{X} and assign each \mathbf{y}_i to the nearest subspace based on SRC [23]. For each out-of-sample data point \mathbf{y}_i , the following optimization problem is solved

$$\min \|\mathbf{c}_i\|_1 \quad \text{s.t.} \quad \|\mathbf{y}_i - \mathbf{X}\mathbf{c}_i\|_2 < \epsilon, \quad (4)$$

where $\epsilon > 0$ is the error tolerance, \mathbf{y}_i denotes an out-of-sample datum and \mathbf{X} denotes in-sample data.

Once the optimal \mathbf{c}_i is obtained, \mathbf{y}_i is assigned to the nearest subspace which has the minimum residual by solving

$$r_j(\mathbf{y}_i) = \|\mathbf{y}_i - \mathbf{X}\delta_j(\mathbf{c}_i)\|_2. \quad (5)$$

$$f(\mathbf{y}_i) = \underset{j}{\operatorname{argmin}}\{r_j(\mathbf{y}_i)\}, \quad (6)$$

where the $f(\mathbf{y}_i)$ denotes the assignment of \mathbf{y}_i , and the nonzero entries of $\delta_j(\mathbf{c}_i) \in \mathbb{R}^p$ are the elements in \mathbf{c}_i associating with the j -th subspace.

Although SRC has achieved a lot of successes in pattern recognition, some recent works [37] showed that non-sparse representation can achieve comparable results with less time cost. Therefore, we perform linear coding scheme instead of sparse one by solving

$$\min_{\mathbf{c}_i} \|\mathbf{y}_i - \mathbf{X}\mathbf{c}_i\|_2^2 + \gamma \|\mathbf{c}_i\|_2^2, \quad (7)$$

where $\gamma > 0$ is a positive real number. The second term is used to avoid over-fitting. Zhang et al. [37] named this method as collaborative representation-based classification (CRC) and empirically showed that collaborative representation rather than the sparse one plays an important role in face recognition. After getting the coefficient of \mathbf{y}_i via solving (7), \mathbf{y}_i is assigned to the subspace that produces the minimal regularized residuals over all classes. Note that, (7) is also known as linear regression based classification [38] when $\gamma = 0$.

Under our framework, SSSC, SLRR, and SLSR are proposed, which make SSC [8], [9], LRR [14], [15], and LSR [19] feasible to cluster large scale and out-of-sample data. Algorithm 2 summarizes our approaches and Fig. 2 gives a toy example to show the effectiveness of our framework. In the example, we use the NodeXL software (a toolkit of Office) [39] to obtain the visualization of the similarity graphs (see Fig. 2(c) and Fig. 2(d)).

B. Error Analysis

In this section, we perform error analysis for the framework. Lemma 1 shows that the clustering partitions solely based on in-sample data $\mathbf{X} \in \mathbb{R}^{m \times p}$ will converge to the partitions based on the whole data set $\mathbf{D} \in \mathbb{R}^{m \times n}$, when $n \rightarrow \infty$ and the sampled data is enough. Based on Lemma 1, we show that the error bound of our framework only depend on the grouping errors of out-of-sample data $\mathbf{Y} \in \mathbb{R}^{m \times (n-p)}$. Moreover, Lemma 2 is the preliminary step to our result.

Lemma 1 ([40]). *Under Assumption 1, if the first k eigenvalues of $\mathbf{L}_{\mathbf{D}}$ have multiplicity 1, then the same holds for the first k eigenvalues of $\mathbf{L}_{\mathbf{X}}$ for sufficiently large p , where $\mathbf{L}_{\mathbf{D}}$ and $\mathbf{L}_{\mathbf{X}}$ denote the Laplacian matrix based on \mathbf{D} and \mathbf{X} , respectively. In this case, the first k eigenvalues of $\mathbf{L}_{\mathbf{X}}$ converge to the first k eigenvalues of $\mathbf{L}_{\mathbf{D}}$, and the corresponding eigenvectors converge almost surely. The clustering partitions constructed by normalized spectral clustering from the first k eigenvectors on finite samples converge almost surely to a limit partition of the whole data space.*

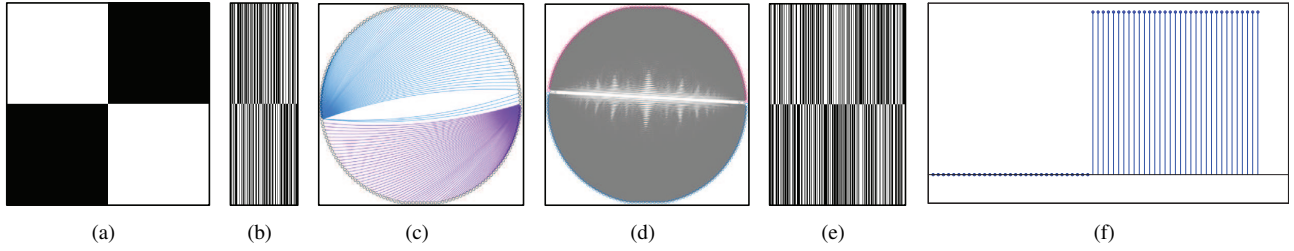


Fig. 2. A toy example based on SSSC and SLRR for showing the effectiveness of our framework. (a) A given data set \mathbf{D} satisfying the sparsity assumption, where the rank of the data equals two; (b) in-sample data \mathbf{X} identifying using unique random sampling method. \mathbf{X} and \mathbf{D} are i.i.d.; (c) The similarity graph of \mathbf{X} achieved by SSSC; (d) The similarity graph of \mathbf{X} achieved by SLRR; (e) out-of-sample data \mathbf{Y} locating in the union of subspaces spanned by \mathbf{X} ; (f) The projection coefficients of an out-of-sample data point $\mathbf{y} \in S_2$, of which only the coefficients over S_2 are nonzero. \mathbf{y} is grouped into the subspace S_2 in terms of our method, which matches with the ground truth. Under Assumption 1, this example shows that our framework can solve the large-scale and the out-of-sample problems for representation-based subspace clustering without loss of clustering quality.

Algorithm 2 Scalable Sparse Subspace Clustering (SSSC), Scalable Low Rank Representation (SLRR), and Scalable Least Square Regression (SLSR).

Input: A given data set $\mathbf{D} \in \mathbb{R}^{m \times n}$, the desired number of clusters k , and the rigid regression parameter $\gamma = 10^{-6}$.

- 1: Randomly select p data points from \mathbf{D} as in-sample data $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p)$. The remaining samples are used as out-of-sample data $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n-p})$.
- 2: Perform SSC or LRR or LSR (Algorithm 1) over \mathbf{X} to get the cluster membership of \mathbf{X} .
- 3: Project each out-of-sample data point \mathbf{y}_i into the union of the subspaces spanned by \mathbf{X} via solving

$$\mathbf{c}_i^* = (\mathbf{X}^T \mathbf{X} + \gamma \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}_i. \quad (8)$$

- 4: Calculate the residuals of \mathbf{y}_i over the j -th subspace by

$$r_j(\mathbf{y}_i) = \|\mathbf{y}_i - \mathbf{X} \delta_j(\mathbf{c}_i^*)\|_2. \quad (9)$$

or the regularized residuals of \mathbf{y}_i over all subspaces via solving

$$r_j(\mathbf{y}_i) = \frac{\|\mathbf{y}_i - \mathbf{X} \delta_j(\mathbf{c}_i^*)\|_2}{\|\delta_j(\mathbf{c}_i^*)\|_2}. \quad (10)$$

- 5: Assign \mathbf{y}_i to the subspace which has the minimal residual by

$$f(\mathbf{y}_i) = \underset{j}{\operatorname{argmin}} \{r_j(\mathbf{y}_i)\}. \quad (11)$$

Output: The cluster membership of \mathbf{D} .

From Lemma 1, we can find that the additive clustering error induced by our framework comes from the process of grouping out-of-sample data \mathbf{Y} . Thus, the problem becomes finding the error boundary of the Nearest Subspace (NS) classifier ((9) or (10)).

The representation-based NS classifiers have been extensively studied in [23], [37], [41], however, theoretical analysis on it receives little attention. [42] presents a theoretical explanation to SRC [23] from the view of maximizing performance margin. However, the error boundary of SRC is still unknown. In this paper, we mainly investigate the performance of SRC (i.e., (9)) from theoretical perspective. To the best of our

knowledge, this is the first work to analyze the error bounds for the NS classifiers.

It is challenging to perform error analysis on the NS classifiers because the active sets (the nonzero set of \mathbf{c}) of different data points are different. In other words, it is difficult to find an invariant set of support vectors to represent each subspace. Therefore, the classic margin analysis theory cannot be directly used to the NS classifiers. To solve this problem, we propose treating each subspace as a point in a hyperspace. We have the following definition.

Definition 1. The hyperspace $\mathcal{H} = \{S, \mathbf{y}\}$ is a set of subspaces, in which each subspace S_j corresponds to a point and the distance between \mathbf{y}_i and S_j is defined as the residual $r_j(\mathbf{y}_i)$.

Based on the above definition, the NS classifier could be regarded as the nearest neighbor classifier in the hyperspace (see Fig. 3) so that one can avoid to find the support vectors for each category. Note that, [43] treats each subspace as a data point in the Grassmann space in which the distance is defined as the principle angle between the subspaces. Clearly, the adopted distance metric is the major difference between Grassmann space and the above defined hyperspace. Indeed, Grassmann space can be regarded as a special case of the hyperspace, which will be further discussed at the end of this section.

Lemma 2 (Cover-Hart inequality [44]). For any distribution of $(\mathbf{Y}, g(\mathbf{Y}))$, the asymptotic error R of the nearest neighbor classifier is bounded by

$$R^* \leq R \leq R^* \left(2 - \frac{k}{k-1} R^* \right), \quad (12)$$

where $g(\mathbf{Y})$ is the ground truth for \mathbf{Y} , k denotes the number of subject, and R^* denotes the Bayes error which is the lowest possible error rate for a given class of classifier.

Based on Lemma 2, the problem is equivalent to estimating the Bayes error in the defined hyperspace. Without loss of generality, we deal with the case of binary classification, i.e., $k = 2$ and $f(\mathbf{y}) = \{-1, 1\}$.

Lemma 3. The error bound of the nearest subspace classifier

$$f(\mathbf{y}_i) = \underset{j}{\operatorname{argmin}} \{\|\mathbf{y}_i - \mathbf{X} \delta_j(\mathbf{c}_i^*)\|\}, \quad (13)$$

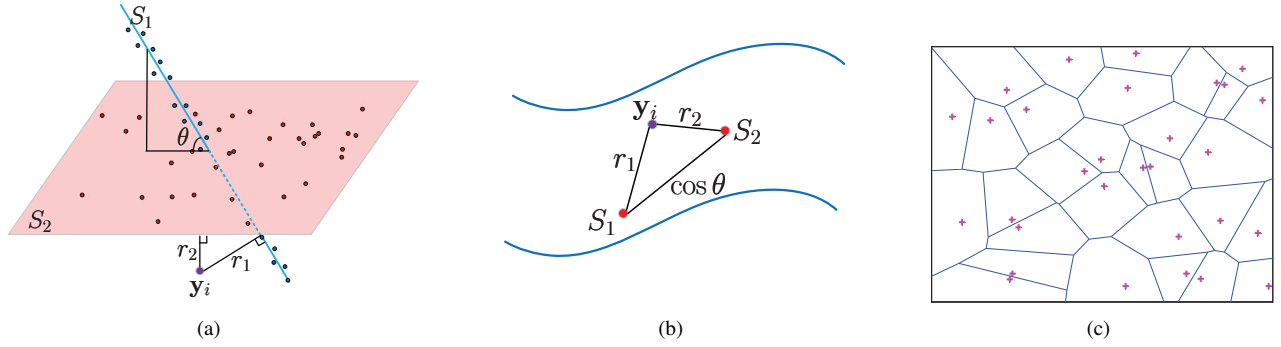


Fig. 3. (a) Two subspaces S_1 and S_2 spanned by in-sample data. We denote an out-of-sample data point by \mathbf{y}_i . θ is the principal angle between S_1 and S_2 , and r_1 and r_2 are the residuals associating with S_1 and S_2 . (b) The hyperspace in which S_1 and S_2 are regarded as two data points. (c) the decision boundary of the nearest subspace classifier in the hyperspace.

is given by

$$\frac{|1 - \max(\alpha_{-1}, \alpha_1)|}{2 + \alpha_{-1} + \alpha_1} \leq R \leq \min\left(0.5, \frac{2 + 2 \min(\alpha_{-1}, \alpha_1)}{|1 - \alpha_{-1}| + |1 - \alpha_1|}\right), \quad (14)$$

where $\mathbf{y}_i \in \mathbb{R}^m$ is the input, $\mathbf{c}_i^* = \Theta \mathbf{y}_i$, $\alpha_j = \|[\mathbf{X}]_j \Theta\|_F$, $[\mathbf{X}]_j \in \mathbb{R}^{m \times p}$ replaces the elements of \mathbf{X} with zeros unless the elements belong to S_j , $j = \{1, 2\}$ denotes the index of subject, $\Theta = (\mathbf{X}^T \mathbf{X} + \gamma \mathbf{I})^{-1} \mathbf{X}^T$, and the nonzero entries of $\delta_j(\mathbf{c}_i^*) \in \mathbb{R}^p$ are the elements in \mathbf{c}_i^* associated with the subspace S_j .

Proof. Let $\eta(\mathbf{y}_i)$ be the conditional probability that the prediction for a given \mathbf{y}_i is 1, i.e., $\eta(\mathbf{y}_i) \triangleq p(f(\mathbf{y}_i) = 1 | \mathbf{y}_i)$. In this case, the Bayes error R^* for \mathbf{y}_i is given by

$$R^*(\mathbf{y}_i) = \min\{\eta(\mathbf{y}_i), 1 - \eta(\mathbf{y}_i)\} \quad (15)$$

According to (15), it is obvious that $0 \leq R^*(\mathbf{y}_i) \leq 0.5$.

We define the probability that \mathbf{y}_i belongs to the subspace S_j using the residual $r_j(\mathbf{y}_i) = \|\mathbf{y}_i - \mathbf{X} \delta_j(\mathbf{c}_i^*)\|_2$, i.e.,

$$\eta(\mathbf{y}_i) = 1 - \frac{r_1(\mathbf{y}_i)}{\sum r_j(\mathbf{y}_i)}. \quad (16)$$

Let $\delta_j(\mathbf{c}_i^*) = \Delta_j \mathbf{c}_i^*$, where $\Delta_j \in \mathbb{R}^{p \times p}$ is a diagonal matrix of which nonzero diagonal entries indicate the columns of \mathbf{X} belonging to the subspace S_j . Since $\mathbf{c}_i^* = \Theta \mathbf{y}_i$, we have

$$r_j(\mathbf{y}_i) = \|\mathbf{y}_i - [\mathbf{X}]_j \Theta \mathbf{y}_i\|_2, \quad (17)$$

where $[\mathbf{X}]_j = \mathbf{X} \Delta_j$.

Thus, to find the bound of (13), we only need to identify the lower and upper bounds of $r_j(\mathbf{y}_i)$.

Step 1: From the reverse triangle inequality of vector norm, we have

$$r_j(\mathbf{y}_i) \geq \|\mathbf{y}_i\|_2 - \|[\mathbf{X}]_j \Theta \mathbf{y}_i\|_2. \quad (18)$$

For any vectors \mathbf{x} and \mathbf{y} , Cauchy-Schwarz inequality suggests that $\|\mathbf{x}^T \mathbf{y}\|_2 \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$. Since the Frobenius norm is subordinate to ℓ_2 -norm, (18) gives that

$$\begin{aligned} r_j(\mathbf{y}_i) &\geq \|\mathbf{y}_i\|_2 - \|[\mathbf{X}]_j \Theta\|_F \|\mathbf{y}_i\|_2 \\ &= \left|1 - \|[\mathbf{X}]_j \Theta\|_F\right| \|\mathbf{y}_i\|_2, \end{aligned} \quad (19)$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Step 2: For any vectors \mathbf{x} and \mathbf{y} , it must hold that $\|\mathbf{x} - \mathbf{y}\|_2 \leq \|\mathbf{x} + \mathbf{y}\|_2 \leq \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2$. Thus, we have

$$\begin{aligned} r_j(\mathbf{y}_i) &= \|\mathbf{y}_i - [\mathbf{X}]_j \Theta \mathbf{y}_i\|_2 \\ &\leq \|\mathbf{y}_i\|_2 + \|[\mathbf{X}]_j \Theta \mathbf{y}_i\|_2 \\ &\leq \|\mathbf{y}_i\|_2 + \|[\mathbf{X}]_j \Theta\|_F \|\mathbf{y}_i\|_2 \\ &= (1 + \|[\mathbf{X}]_j \Theta\|_F) \|\mathbf{y}_i\|_2 \end{aligned} \quad (20)$$

Let $\alpha_j = \|[\mathbf{X}]_j \Theta\|_F^2$ and combine (16), (19), and (20), we have

$$\frac{|1 - \alpha_{-1}|}{2 + \alpha_{-1} + \alpha_1} \leq \eta(\mathbf{y}_i) \leq \frac{1 + \alpha_{-1}}{|1 - \alpha_{-1}| + |1 - \alpha_1|}, \quad (21)$$

and

$$\frac{|1 - \alpha_1|}{2 + \alpha_{-1} + \alpha_1} \leq 1 - \eta(\mathbf{y}_i) \leq \frac{1 + \alpha_1}{|1 - \alpha_{-1}| + |1 - \alpha_1|}, \quad (22)$$

respectively.

Clearly, the boundary of the expected Bayes error $R^* = E\{R^*(\mathbf{Y})\}$ is independent of the out-of-sample data \mathbf{Y} . From Lemma 2, the following relations hold:

$$R^* \leq R \leq 2R^* (1 - R^*) \leq 2R^*. \quad (23)$$

Since $0 \leq R^* \leq 0.5$, then from (21), (22), and (23), we have

$$\frac{|1 - \max(\alpha_{-1}, \alpha_1)|}{2 + \alpha_{-1} + \alpha_1} \leq R \leq \min\left(0.5, \frac{2 + 2 \min(\alpha_{-1}, \alpha_1)}{|1 - \alpha_{-1}| + |1 - \alpha_1|}\right). \quad (24)$$

This completes the proof. \square

From the above analysis, we can conclude that:

- The error bound only depends on the structure of the subspaces spanned by in-sample data under Assumption 1. Indeed, the structure of the subspaces is also the unique factor to affect the clustering quality as shown in [9], [15]. Thus, we argue that our framework solves the large-scale and the out-of-sample problems for the representation-based subspace clustering methods without introducing new error factors. This is largely different from the traditional methods [30], [45] whose performance depends on the sampling rate.

²In practice, we often normalize α_j via $\alpha_j = \alpha_j / \sum_j \alpha_j$.

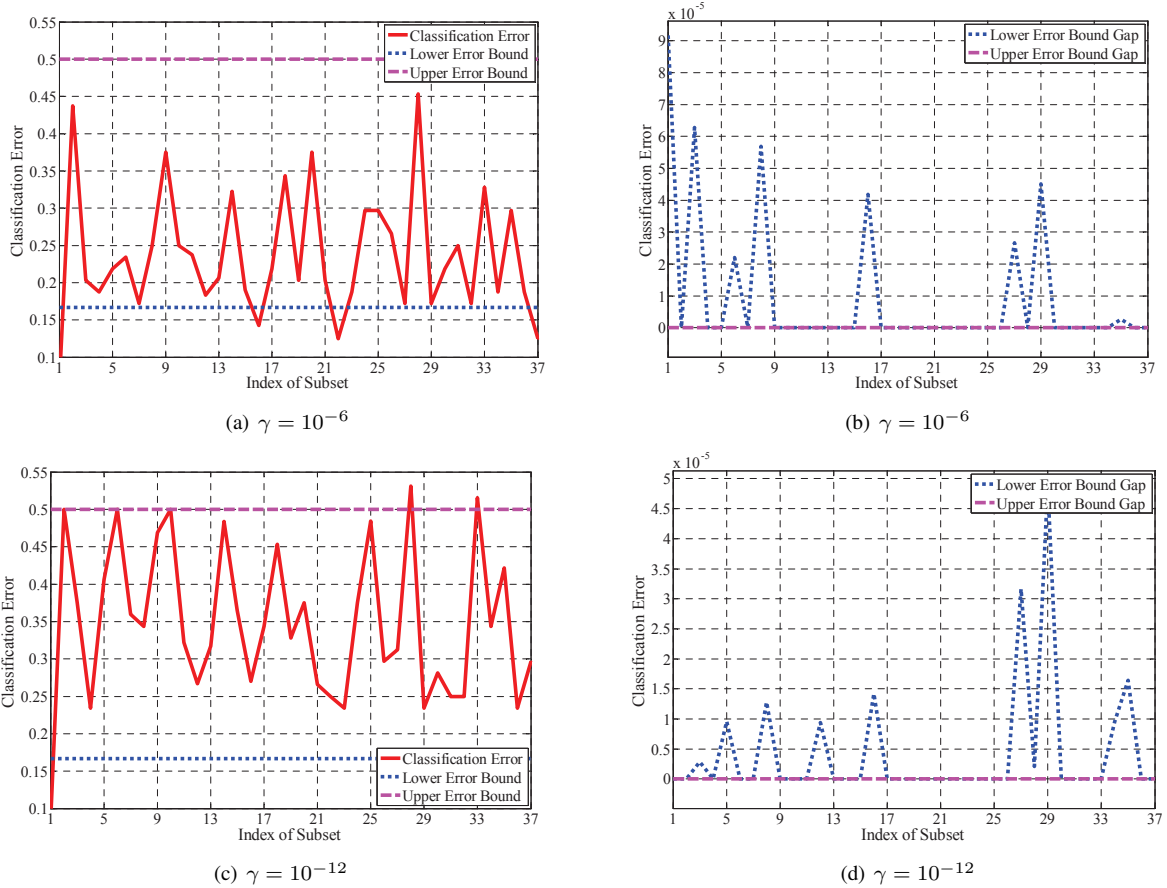


Fig. 4. A real-world example to validate the estimated error bounds. (a) and (c): Classification error and error bound of eq.(13) on 37 subsets of Extended Yale database B. Each subset consists of the samples from the first category and another category. (b) and (d): The gap between two different error bounds derived from equations (22) and (25).

- Considering $\mathbf{X}^T \mathbf{X}$ is well conditioned, then one sets $\gamma = 0$. $\alpha_j = [\mathbf{X}]_j \mathbf{X}$ measures the similarity between the subspace $[\mathbf{X}]_j$ and \mathbf{X} using their inner product. More generally (*i.e.*, $\gamma > 0$), let θ_i be the i -th principal angle between $[\mathbf{X}]_j$ and Θ , then, it holds that $\sigma_i = \cos \theta_i$, where σ_i is the i -th singular value of $[\mathbf{X}]_j \Theta$. According to the definition of the Frobenius norm, *i.e.*, $\alpha_j = \sqrt{\sum_i^{r_j} \sigma_i^2}$, we have $\alpha_j = \|[\mathbf{X}]_j \Theta\|_F = \sqrt{\sum_i^{r_j} \cos^2 \theta_i}$ which measures the distance between $[\mathbf{X}]_j$ and Θ by their principal angles, where r_j is the rank of $[\mathbf{X}]_j \Theta$.

Under Assumption 1, our error analysis method is validate only when the following two conditions are satisfied when: 1) the data are sampled from two subspaces, *i.e.*, $k = 2$. If $k > 2$, one may extend our method by recursively transforming the multiple clusters problem into binary one even though this task may need massive effort; and 2) in-sample data $[\mathbf{X}]_j$ have been correctly clustered. Otherwise, one needs identify the error bound for the whole framework not just for grouping out-of-sample data. The difficulty of this task is how to identify the influence of perturbation due to sampling. A possible way to solve this problem is perturbation theory that has been studied in quantum mechanics. However, this is beyond the main scope of this paper.

To validate our theoretical results, we perform experiments

on 37 subsets of Extended Yale database B [46]. Each subset consists of the samples from the first category and one of the others. We use 64 (32 samples per subject) samples for training and the remaining samples for testing. Moreover, we use principle components analysis (PCA) as the preprocess step to extract 60 features from training and testing data. Fig. 4 shows results from which one can find that:

- We successfully estimate the error bounds for 33 and 34 out of 37 subsets in the case of $\gamma = 10^{-6}$ and $\gamma = 10^{-12}$, respectively. The failure cases may be attributed to the following reasons: First, the classification error (solid line) is calculated based on training data and testing data, whereas the error bounds (dotted lines) are estimated only based on training data. When training data cannot represent the distribution of the whole data space, the estimated bounds will be incorrect. Second, our analysis is based on Assumption 1, which may not be perfectly satisfied by real-world data (*e.g.*, the Extended Yale database B) since real-world data is often complex.
- Figs. 4(a) and 4(c) show that a larger γ may reduce the classification error rate, while increasing the failure rate of our error analysis method. The reason is that γ is used to avoid overfitting by adding a value to the diagonal entries of $\mathbf{X}^T \mathbf{X}$, which actually affects the structure of the observed data.

TABLE II
 COMPUTATIONAL COMPLEXITY OF SSC, LRR, LSR, AND THEIR
 SCALABLE VERSIONS PROPOSED IN THIS PAPER. t_1 , t_2 , AND t_3
 CORRESPOND TO THE NUMBER OF ITERATIONS OF THE ℓ_1 -SOLVER, THE
 RANK-MINIMIZER, AND THE K-MEANS CLUSTERING METHOD,
 RESPECTIVELY.

Algorithms	Time Complexity	Space Complexity
SSC [8], [9]	$t_1 mn^3 + t_2 nk^2$	mn^2
SSSC	$t_1 mp^3 + t_2 pk^2 + np^2$	mp^2
LRR [14], [15]	$t_3(m^2n + n^3) + t_2 nk^2$	mn^2
SLRR	$t_3(pm^2 + p^3) + t_2 p^3 + np^2$	mp^2
LSR [19]	$m^2n + n^3 + t_2 nk$	mn^2
SLSR	$pm^2 + np^2 + t_2 pk$	mp^2

- Besides (22), we derive another bound for $1 - \eta(\mathbf{y}_i)$ from (21) instead of (16), (19), and (20), *i.e.*,

$$1 - \frac{1 + \alpha_{-1}}{|1 - \alpha_{-1}| + |1 - \alpha_1|} \leq 1 - \eta(\mathbf{y}_i) \leq 1 - \frac{|1 - \alpha_{-1}|}{2 + \alpha_{-1} + \alpha_1} \quad (25)$$

Figs. 4(b) and 4(d) shows the gap between these two different formulations. Clearly, the gaps are close to zero.

C. Complexity Analysis

Suppose p samples are selected from n data points with dimensionality of m , SSSC needs $O(t_1 p^2 m^2 + t_1 m p^3 + p^2 + t_2 p k^2)$ to get the cluster membership of in-sample data when the Homotopy optimizer [47] is used to solve the ℓ_1 -minimization problem and the Lanczos eigensolver is used to compute the eigenvectors of $\mathbf{L} \in \mathbb{R}^{p \times p}$, where k is the number of clusters, and t_1 and t_2 is the number of iterations of Homotopy optimizer and the k-means algorithm, respectively. To group out-of-sample data points, SSSC needs to compute the pseudo-inverse of the an $m \times m$ matrix and calculate the linear representation of $\mathbf{Y} \in \mathbb{R}^{m \times (n-p)}$ in $O(pm^2 + p^3 + (n-p)p^2)$.

Putting everything together, the computational complexity of SSSC is $O(t_1 m p^3 + t_2 p k^2 + np^2)$ since $k, m < p \ll n$. Clearly, the cost of SSSC is largely less than that of SSC ($O(t_1 m n^3 + t_2 n k^2)$). In the similar way, one can get the computational complexities of SLRR and SLSR. Table II reports the computational complexities of our methods and the original algorithms.

IV. EXPERIMENTAL RESULTS

In this section, we carry out some experiments to show the effectiveness and efficiency of SSSC, SLRR, and SLSR.

The experiments consist of five parts, Section IV-C investigates the performance of our methods to the varying parameters; Section IV-D reports the results of all the evaluated algorithms with different sampling rates; Section IV-E compares our methods with the corresponding original algorithms on three facial data sets. Moreover, we also investigate the performance of two nearest subspace classifiers (9) and (10); Section IV-F reports the clustering quality of the tested methods on three medium-sized data sets including facial images, handwritten digital data, and documental corpus; Section IV-G shows the results on three large scale data sets.

A. Data Sets

We perform experiments on nine real-world data sets including facial images, handwritten digital data, news corpus, etc. The data sets consist of three small-sized data sets, three medium-sized data sets, and three large scale data sets. We presented some statistics of the data sets in Table III and a brief description as follows.

In general, facial images are assumed to be located in the low-dimensional manifold. In the experiments, we investigate four popular facial data sets, *i.e.*, AR [48], Extended Yale database B (ExYaleB) [46], Labeled Faces in the Wild-a (LFW) [49], and Multi-PIE (MPIE) [50]. AR includes over 4,000 face images of 126 people (70 male and 56 female). In our implementation, we used a subset of AR which contains 1,400 clean faces randomly selected from 50 male subjects and 50 female subjects. LFW contains 13,123 images captured from uncontrolled environment with variations of pose, illumination, expression, misalignment, and occlusion. We use a subset of the aligned LFW which includes 143 subjects with no less than 11 samples per subject. MPIE contains the facial images of 286 individuals captured in four sessions with simultaneous variations in pose, expression and illumination³. We use all frontal images from all the sessions. For computational efficiency, we downsize 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). Moreover, we perform PCA over the downsized data to retain 98% energy. For each LFW image, “divide and conquer” strategy is adopted as did in [51]. In details, each image is partitioned into 2×2 blocks; and then the discrimination-enhanced feature in each block is extracted; after that, all blocks’ features are concatenated to form the final feature vector.

Reuters-21578 (RCV) [52] is a documental corpus. In the experiments, the first 785 principle components of RCV are extracted as the features. We also use three UCI data sets⁴, *i.e.*, PenDigits, Covtype [53], and PokerHand [54]. PokerHand is an unbalanced data set, of which the maximal class contains 501,209 samples, compared with 3 samples of the minimal class. We examine the performance of the tested algorithms using the original data set (PokerHand-2) and a subset (PokerHand-1) with 971,329 data points from three largest subjects.

B. Baseline Algorithms and Evaluation Metrics

Spectral clustering and kernel-based clustering methods are popular to cope with linearly inseparable data. Some studies [56] have established the equivalence between them. In the experiments, we compare the proposed methods with four scalable spectral clustering algorithms (KASP [30], Nyström approximation based spectral clustering [28], [29], LSC [31], and SEC [33]) and one scalable kernel-based clustering approach (AKK [45]). Moreover, we report the results of the k-means clustering algorithm [57] as a baseline. Besides our

³illuminations of the used MPIE: 0,1,3,4,6,7,8,11,13,14,16,17,18,19

⁴<http://archive.ics.uci.edu/ml/datasets.html>

TABLE III
DATA SETS USED IN THE EXPERIMENTS. THE NUMBER IN THE PARENTHESES DENOTES THE RETAINING ENERGY BY PCA.

Data sets	# samples	Dim.	# features	# classes
AR [48]	1,400	19,800	167 (98%)	100
ExYaleB [46]	2,414	32,256	114 (98%)	38
LFW [55]	4,174	62,500	560	143
MPIE [50]	8,916	8,200	115 (98%)	286
RCV [52]	8,293	18,933	785 (85%)	65
PenDigits	10,992	16	16	10
Covtype [53]	581,012	54	54	7
PokerHand-1 [54]	971,329	10	10	3
PokerHand-2 [54]	1,000,000	10	10	10

own implementation, we also quote some results directly from the literature.

We investigate 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-Orth is orthogonal, whereas that of Nyström is not. SEC obtains the results by performing k-means in the embedding space. All algorithms are implemented in MATLAB. The used data sets and the codes of our algorithms can be downloaded at www.machineilab.org/users/pengxi/.

The evaluated algorithms take two approaches to find in-sample data. Specifically, SSSC, SLRR, SLSR, Nyström, Nyström_Orth, LSC_R, SEC and AKK identify in-sample data by performing uniform random sampling method, whereas KASP and LSC_K adopt the k-means clustering method. To avoid the disparity in data partitions, we pre-partition each data set into two parts, in-sample data and out-of-sample data. After that, we run different algorithms run over these data partitions.

We measure the clustering quality using *Accuracy* [58] and Normalized Mutual Information (*NMI*) [52] between the produced clusters and the ground truth categories. The *Accuracy* or *NMI* of 1 indicates perfect matching with the true subspace distribution, whereas 0 indicates totally mismatch.

To be consistent with the previous works [9], [15], we tune the parameters of all the evaluated methods to achieve the highest *Accuracy*. For SSSC, we adopted the Homotopy optimizer to solve the ℓ_1 -minimization problem. The optimizer has two user-specified parameters, sparsity parameter λ and error tolerance parameter δ . We tuned the parameters in the range of $\lambda = (10^{-7}, 10^{-6}, 10^{-5})$ and $\delta = (10^{-3}, 10^{-2}, 10^{-1})$. For SLRR and SLSR, the value of λ is chosen as shown in Fig. 5. Referring to the parameter setting in [29], [30], [31], [33], [45], the parameter τ of KASP and Nyström was set as [0.1, 1] with an interval of 0.1 and [2, 20] with an interval of 1; the parameter σ of AKK ranges from [0.1, 1] with an interval of 0.1; SEC has three user-specified parameters, *i.e.*, the size of neighborhood r , balanced parameters μ and γ . We set $\gamma = 1$, $\mu = [10^{-9}, 10^{-6}, 10^{-3}, 10^0, 10^{+3}, 10^{+6}, 10^{+9}, 10^{+12}, 10^{+15}]$, and r from 2 to 20. Moreover, the same value range of r was used for KASP and LSC.

Following the common benchmarking procedures, we run each algorithm five times on each data set and report the final results by the mean and standard deviation of the *Accuracy*

(*NMI*) and the mean of time costs.

C. The Influence of Parameters

SSSC uses $\lambda > 0$ to control the sparsity of the representation and $\epsilon > 0$ to measure the reconstruction errors. SLRR uses $\lambda > 0$ to balance different parts in the objective function and SLSR utilizes $\lambda > 0$ to avoid overfitting. The choice of these parameters depends on the data distribution.

Fig. 5 shows the results of SSSC, SLRR, and SLSR with different parameter values. When λ or ϵ of SSSC is assigned with a small positive value (from 10^{-7} to 0.01), it achieves a good performance. When the parameters are assigned with a big value, the performance of SSSC is degraded. For SLRR, while λ ranges from 0.5 to 3.9, its *Accuracy* and *NMI* are almost unchanged. SLSR performs worse with increasing λ . This verifies our claim that a small λ is preferable to the clean data set.

D. The Influence of In-sample Data Size

To study the influences of in-sample data size p , we perform experiments on ExYaleB 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. 6 reports the result, from which we have the following observations:

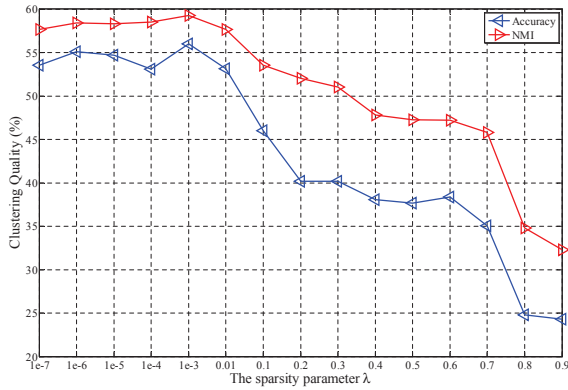
- Except SEC and AKK, all the scalable clustering methods outperform the k-means method in *Accuracy* and *NMI*. SSSC, SLRR, and SLSR are superior to the other investigated approaches by a considerable performance margin. For example, SLRR achieves 15.1% gain in *Accuracy* and 13.1% gain in *NMI* over the best baseline algorithm (Nyström) when $p = 228$.
- In most cases, all the algorithms except Nyström and Nyström_Orth perform better with increasing p . The possible reason for this result is that Nyström and Nyström_Orth speed up the clustering process by reducing the size of affinity matrix rather than data size.
- The accuracy of SLRR decreased when p increased from 912 to 1368. The result seems inconsistent with the common sense that more data tend to bring better performance. This result can be attributed to the characteristic of SLRR, *i.e.*, SLRR is based on low rank representation that incorporates the relations among different subspaces. Increasing p would result in more intersections among different subspaces and weaken the discrimination of model. To obtain an optimal p , some model selection methods such as M -estimator [59] could be used.

E. Clustering on Small Scale Data

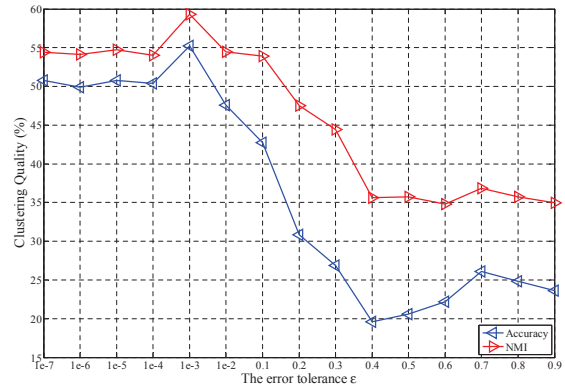
We carry out the experiments on three facial data sets, *i.e.*, AR, ExYaleB, and LFW. Moreover, we investigate the performance of our methods when the classifiers (9) and (10) are used to group out-of-sample data. In the experiments, we fix $\epsilon = 10^{-3}$ for SSSC and SSC.

From Table IV, we can find that

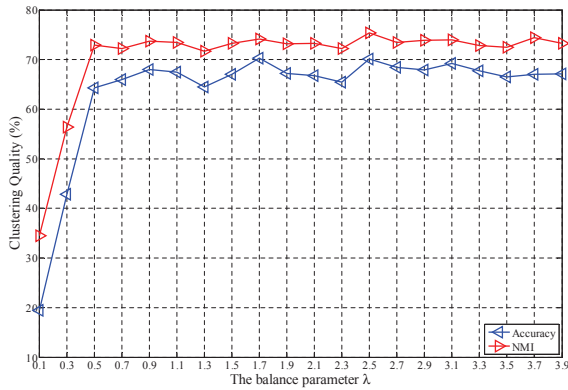
- Our framework successfully makes SSC, LRR, and LSR feasible to group out-of-sample data with acceptable loss



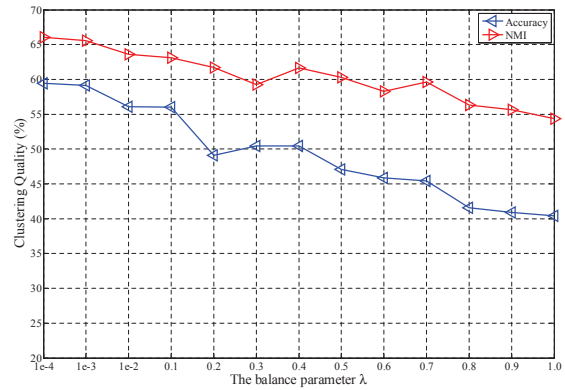
(a) The influences of the parameter λ of SSSC, where $\delta = 10^{-3}$.



(b) The influences of the parameter δ of SSSC, where $\lambda = 10^{-3}$.



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



(d) The influences of the parameter λ of SLSR.

Fig. 5. The influence of the parameters. A half of images (1212) are chosen from ExYaleB as in-sample data and the rest are used as out-of-sample data. The x-coordinate denotes the values of the parameters, and the y-coordinate corresponds to the clustering quality (*Accuracy* and *NMI*).

in clustering quality. For example, the *Accuracy* of SSC on AR data set is 9.73% higher than that of SSSC, whereas the time cost of SSC is about three times that of SSSC. With the increase of data size, SSC, LRR, and LSR will fail to get the results, whereas SSSC, SLRR, SLSR can get the results with an acceptable time cost.

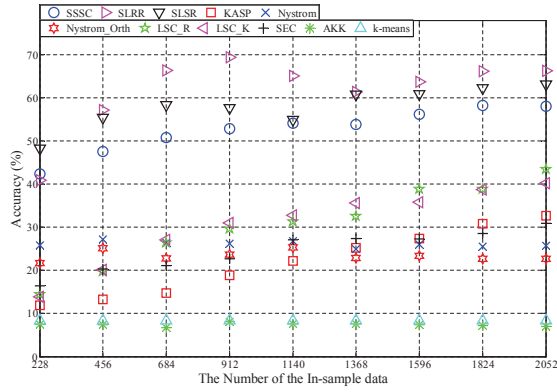
- Compared with the other scalable methods (*i.e.*, KASP, Nyström, Nyström_Orth, LSC_R, LSC_K, SEC, and AKK), SSSC, SLRR, and SLSR find an elegant balance between the clustering quality and the time costs. Although SSSC, SLRR, and SLSR are not the fastest, they achieve the best results.
- SLRR performs better than SSSC in the tests. The possible reason is that the low rank representation could capture the structure among different categories, whereas sparse representation cannot, as pointed out in [21]. Moreover, the regularized residual based classifier (10) perform slightly better than the non-regularized residual based classification method (9).
- Nie *et al.* [33] investigated the performance of SEC on ExYaleB. The highest *Accuracy* of SEC is about 42.8% in their tests, comparing with 22.02% in our experiment. The potential reason for the performance difference is that they adopted spectral rotation to get the cluster membership, whereas we use the k-means clustering method. Note that, the best result (42.8%) of SEC reported in

their work is remarkably lower than the results achieved by SSSC (55.5 ± 1.26), SLRR (68.9 ± 1.19), and SLSR (58.9 ± 1.45).

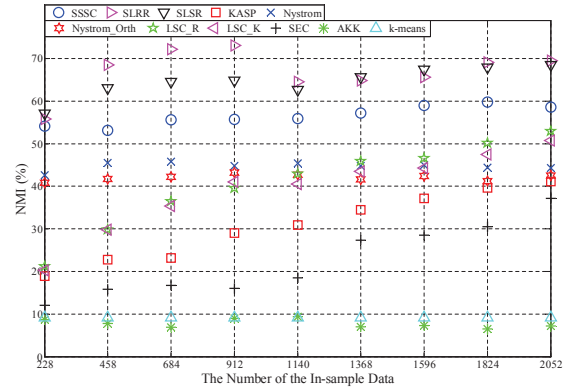
F. Clustering on Medium Scale Data

This section investigates the performance of our methods on MPIE (facial images), RCV (documental corpus), and PenDigits (handwritten digital data). The tuned ϵ of SSSC are 10^{-4} , 0.1, and 0.01, respectively. Table V reports the clustering quality and the time cost (seconds) of the tested methods, from which we can find that

- Our methods outperform the other scalable methods. For example, SLRR achieves a 10.4% gain in *Accuracy* on MPIE over the best competing algorithm (Nyström_Orth), and the gains achieved by SSSC and SLSR are about 7.3% and 8.6%, respectively.
- The running time is a weakness of SSSC, SLRR, and SLSR even though they are more efficient than the original approaches. We have found that most of the time was consumed to handle in-sample data. For example, SSSC takes 840.6 seconds to cluster in-sample data and 220.63 seconds to handle out-of-sample data in the case of RCV. Since in-sample data clustering is an offline process, we assume that our algorithms are more competitive in large scale setting as shown in Section IV-G.



(a) Accuracy versus the varying in-sample data size



(b) NMI versus the varying in-sample data size

Fig. 6. Clustering quality of the competing algorithms on the Extended Yale database B. The x-coordinate denotes in-sample data size and the y-coordinate denotes the clustering quality (*Accuracy* or *NMI*).

TABLE IV
PERFORMANCE COMPARISON (MEAN±STD) AMONG DIFFERENT ALGORITHMS OVER THREE POPULAR FACIAL DATA SETS.

Data sets	AR ($p = 700$)			ExYaleB ($p = 1212$)			LFW ($p = 1000$)		
Algorithm	Accuracy (%)	NMI (%)	Time	Accuracy (%)	NMI (%)	Time	Accuracy (%)	NMI (%)	Time
SSSC	60.4±1.74(10^{-5})	80.8±0.99	142.2	55.5±1.26(10^{-5})	60.3±0.29	128.0	27.6±0.51(10^{-6})	43.7±0.12	184.0
SLRR	70.6±1.50(3.1)	87.3±0.44	40.1	68.9±1.19 (2.9)	74.0±0.60	26.8	30.4±0.46 (0.7)	44.8±0.19	228.4
SLSR	78.7±1.42 (10^{-2})	89.6±0.40	32.4	58.9±1.45(10^{-4})	65.2±0.61	21.4	28.5±0.32(0.7)	43.8±0.32	213.4
KASP [30]	32.5±0.55(0.1)	63.6±0.57	134.8	20.6±1.28(8)	31.3±0.93	37.8	25.1±0.93(7)	42.7±0.40	251.4
Nyström [29]	62.2±1.71(2)	82.1±1.16	2.3	20.7±1.16(12)	39.7±0.63	8.2	26.6±0.81(0.6)	42.0±0.33	3.2
Nyström_Orth	57.5±3.55(0.9)	79.1±1.80	13.7	21.4±1.50(3)	40.3±1.01	60.9	26.7±0.86(0.5)	41.0±0.37	11.3
LSC_R [31]	31.1±0.71(4)	61.3±0.52	1.7	32.3±0.91(2)	43.7±0.34	7.3	25.9±0.48(3)	41.5±0.28	3.9
LSC_K [31]	32.9±0.79(4)	62.9±0.50	2.2	31.2±2.07(2)	42.1±1.33	8.3	22.0±0.50(10)	41.6±0.28	5.1
SEC [33]	25.9±1.81(10^{+9} , 8)	41.1±1.60	1.7	22.0±1.68(10^{-9} , 1)	39.4±1.85	10.3	25.2±1.68(10^{+12} , 4)	40.4±1.49	2.3
AKK [45]	22.0±1.28(0.2)	52.0±1.09	0.8	6.8±0.48(0.4)	5.5±0.82	3.0	16.0±0.99(0.3)	34.7±0.81	2.7
SSSC2	58.3±1.38(10^{-5})	79.6±0.49	79.6	57.8±1.21(10^{-5})	62.3±0.60	65.0	26.5±0.22(10^{-6})	43.2±0.11	212.7
SLRR2	69.1±2.50(3.1)	86.2±0.77	39.8	71.8±0.91(2.9)	77.3±0.45	30.1	29.1±0.49(0.7)	43.9±0.08	321.6
SLSR2	77.6±1.30(10^{-2})	88.7±0.55	30.1	61.2±1.35(10^{-4})	67.3±0.90	23.6	28.8±0.39(0.7)	43.6±0.15	232.4
k-means [57]	29.1±0.59(-)	58.4±0.43	18.8	8.4±0.50(-)	9.9±0.72	50.2	19.4±0.56(-)	37.3±0.27	87.7
SSC [9]	70.1±1.85(10^{-7})	86.4±0.73	361.4	59.0±0.91(10^{-3})	65.1±0.34	344.9	31.6±0.64(10^{-5})	47.5±0.24	804.0
LRR [15]	78.6±0.02(1)	89.3±0.59	152.9	73.7±0.01 (2.1)	78.5±0.46	46.9	36.4±0.02(3.2)	51.3±0.36	623.1
LSR [19]	81.4±1.77 (10^{-2})	91.4±0.60	104.8	68.7±2.11(0.2)	72.9±1.58	89.6	37.9±0.66 (0.9)	54.1±0.23	243.9

Note: p denotes in-sample data size. The number in the parenthesis are the tuned parameters. SSC, LRR, LSR, and the k-means method cannot handle out-of-sample data. Thus, the results of these four methods are achieved by directly performing them on the whole data set. SSSC, SLRR, and SLSR assign out-of-sample data to the nearest subspace which has minimal residual (*i.e.*, eq.(10)), whereas SSSC2, SLRR2, and SLSR2 get the results using eq.(9). The **bold number** indicate the best performance.

- In most cases, LSC_K outperforms LSC_R with a little improvement, which verifies the claim [60] that the complex sampling techniques actually cannot produce a better result than the random sampling method.
- [31] also investigated the *Accuracy* of LSC_R, LSC_K, Nyström_Orth, and KASP on the PenDigits data set. The highest *Accuracy* of these algorithms are 79.0%, 79.3%, 73.9% and 72.5%, which is close to the results achieved in our experiments (*i.e.*, 77.7%, 79.9%, 67.3% and 73.1%).

G. Clustering on Large Scale Data

Table VI reports the performance of our algorithms on 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 assign $\epsilon = 0.2$ to SSSC on Covtype

and PokerHand-2 and fix $\epsilon = 0.1$ in the case of PokerHand-1. We have the following observations:

- SSSC, SLRR, and SLSR outperform the other approaches in all the tests. For example, the *Accuracy* of SSSC is at least 4.7% higher than the other tested methods on Covtype. On PokerHand-1 and PokerHand-2, the gains are 3.7% and 2.1%, respectively.
- The *NMI* achieved by all the tested methods are close to 0. This shows that the metric *NMI* failed to distinct the performance of the evaluated algorithms.
- In [31], the highest *Accuracy* on Covtype achieved by LSC_R, LSC_K, Nyström_Orth and KASP are 24.7%, 25.5%, 22.3% and 22.4%, respectively. In our experiments, the *Accuracy* of these four algorithms are 22.0%, 22.0%, 23.3% and 23.9%, respectively. The possible reason may attribute to the subtle engineering details, *e.g.*, the in-sample and out-of-sample data partitions.

TABLE V
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS ON THREE MEDIUM-SIZED DATA SETS, *i.e.*, MPIE, RCV, AND PENDIGITS.

Data sets	MPIE ($p = 1000$)			PenDigits ($p = 1000$)			RCV ($p = 2000$)		
Algorithm	Accuracy (%)	NMI (%)	Time	Accuracy (%)	NMI (%)	Time	Accuracy (%)	NMI (%)	Time
SSSC	$57.6 \pm 0.97(10^{-6})$	79.4 ± 0.40	432.1	$80.0 \pm 1.31(10^{-7})$	71.3 ± 0.11	17.0	$19.6 \pm 1.34(10^{-7})$	29.8 ± 0.58	840.6
SLRR	$60.7 \pm 0.62(2.30)$	78.9 ± 0.34	340.4	$74.8 \pm 0.92(0.30)$	67.6 ± 0.00	10.4	$49.1 \pm 0.11(3.10)$	31.3 ± 0.26	499.6
SLSR	$59.0 \pm 0.58(0.60)$	79.5 ± 0.49	355.4	$78.4 \pm 0.81(1.00)$	69.6 ± 0.01	8.9	$11.2 \pm 0.41(0.60)$	18.3 ± 1.22	95.2
KASP [30]	$16.6 \pm 0.53(0.1)$	57.0 ± 0.28	1479.8	$73.1 \pm 6.37(4)$	75.5 ± 3.39	12.5	$19.0 \pm 0.64(0.1)$	26.7 ± 0.33	198.8
Nyström [29]	$47.1 \pm 1.46(0.7)$	77.2 ± 0.88	15.3	$66.7 \pm 6.93(0.4)$	65.4 ± 2.70	35.9	$15.9 \pm 1.10(0.4)$	27.7 ± 0.37	27.1
Nyström_Orth	$50.3 \pm 2.38(0.7)$	78.1 ± 1.62	64.8	$67.3 \pm 5.66(3)$	64.8 ± 2.67	6.2	$19.8 \pm 0.53(0.1)$	23.7 ± 0.39	3401.3
LSC_R [31]	$18.1 \pm 0.11(2)$	54.5 ± 0.25	62.1	$77.7 \pm 3.18(15)$	74.9 ± 2.61	5.6	$15.4 \pm 0.15(2)$	22.2 ± 0.15	8.9
LSC_K [31]	$17.5 \pm 0.37(3)$	56.1 ± 0.46	65.7	$79.9 \pm 2.73(11)$	76.4 ± 0.58	7.9	$22.0 \pm 1.83(2)$	34.5 ± 0.43	17.7
SEC [33]	$13.2 \pm 0.39(10^{-3}, 9)$	44.1 ± 0.43	27.2	$75.3 \pm 4.20(10^{-9}, 4)$	70.3 ± 2.43	11.8	$14.8 \pm 0.67(10^{-6}, 3)$	26.3 ± 0.52	19.9
AKK [45]	$10.4 \pm 0.19(0.1)$	38.7 ± 0.66	24.6	$69.0 \pm 4.64(0.01)$	66.9 ± 1.63	6.2	$18.3 \pm 0.62(0.2)$	31.6 ± 0.30	27.9
k-means [57]	$14.5 \pm 0.36(-)$	53.2 ± 0.26	268.5	$77.0 \pm 0.13(-)$	69.2 ± 0.02	23.7	$19.3 \pm 1.10(-)$	23.8 ± 0.52	256.8

TABLE VI
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS OVER THREE LARGE SCALE DATA SETS, *i.e.*, COVTYPE ($n = 581, 012$), POKERHAND-1 ($n = 971, 329$), AND POKERHAND-2 ($n = 1, 000, 000$).

Data sets	Covtype ($p = 1000$)			PokerHand-1 ($p = 1000$)			PokerHand-2 ($p = 1000$)		
Algorithm	Accuracy (%)	NMI (%)	Time	Accuracy (%)	NMI (%)	Time	Accuracy (%)	NMI (%)	Time
SSSC	$28.6 \pm 0.00(10^{-5})$	5.3 ± 0.00	325.5	$51.6 \pm 0.00(10^{-7})$	0.3 ± 0.00	267.7	$17.6 \pm 0.00(10^{-5})$	0.1 ± 0.10	474.1
SLRR	$27.1 \pm 0.03(0.10)$	3.6 ± 0.02	240.9	$37.8 \pm 0.00(0.10)$	0.1 ± 0.00	166.9	$16.0 \pm 0.00(0.10)$	0.1 ± 0.00	317.7
SLSR	$26.5 \pm 0.00(0.01)$	7.2 ± 0.00	268.8	$37.0 \pm 0.00(0.10)$	0.0 ± 0.00	167.8	$15.8 \pm 0.01(0.10)$	0.1 ± 0.00	494.2
KASP [30]	$23.9 \pm 1.93(3)$	3.5 ± 0.19	1314.5	$34.7 \pm 0.93(0.3)$	0.0 ± 0.00	5497.1	$11.3 \pm 0.32(3)$	0.1 ± 0.04	7049.9
Nyström [29]	$24.0 \pm 0.59(0.1)$	3.8 ± 0.03	40.6	$47.9 \pm 0.02(0.2)$	0.2 ± 0.01	61.4	$12.9 \pm 0.27(0.2)$	0.2 ± 0.04	205.7
Nyström_Orth	$23.3 \pm 0.67(0.1)$	3.8 ± 0.16	351.6	$35.8 \pm 0.33(20)$	0.1 ± 0.00	204.4	$15.6 \pm 2.89(17)$	0.1 ± 0.02	205.7
LSC_R [31]	$22.0 \pm 0.47(2)$	3.8 ± 0.06	154.5	$34.9 \pm 0.01(8)$	0.0 ± 0.00	1891.0	$12.6 \pm 0.17(5)$	0.0 ± 0.04	1936.8
LSC_K [31]	$22.0 \pm 0.52(4)$	3.6 ± 0.10	1155.4	$32.4 \pm 1.03(2)$	0.0 ± 0.00	8765.5	$13.8 \pm 0.51(3)$	0.1 ± 0.02	8829.0
SEC [33]	$21.1 \pm 0.01(1, 4)$	3.6 ± 0.00	64.9	$36.6 \pm 0.00(10^{-9}, 3)$	0.1 ± 0.00	81.4	$10.5 \pm 0.06(10^{-3}, 4)$	0.1 ± 0.01	130.2
AKK [45]	$22.8 \pm 1.63(1)$	3.8 ± 0.08	344.2	$35.9 \pm 0.04(0.1)$	0.1 ± 0.00	1039.3	$10.5 \pm 0.06(0.01)$	0.0 ± 0.01	2882.5
k-means [57]	$20.8 \pm 0.00(-)$	3.7 ± 0.00	4895.7	$36.0 \pm 0.01(-)$	0.1 ± 0.00	4760.4	$10.4 \pm 0.06(-)$	0.0 ± 0.01	7188.8

- With the increase of data size, our methods demonstrate a good balance between the running time and the clustering quality. Moreover, the used memory of our methods only depends on in-sample data size, which makes our methods are very competitive in large scale setting.

In summary, we can conclude that the three new methods outperform the competing algorithms in all the tests. In particular, SSSC is more advantageous on large scale data sets (*e.g.*, Covtype and PokerHand), while SLRR outperforms on high-dimensional data clustering problems (*e.g.*, facial images and documental corpus). SLSR can achieve comparable clustering performance with SSSC and SLRR, but has higher computational efficiency than the latter.

V. CONCLUSION

In this paper, we proposed a general framework to solve the large-scale and the out-of-sample clustering problems for representation-based subspace clustering. Under our framework, we further presented three scalable methods, *i.e.*, SSSC, SLRR, and SLSR, which largely reduce the computational complexity of the original methods while preserving a good performance. We proved that the performance of our method only depends on the latent structure of the data set and is independent of the sampling rate. Moreover, we proposed a novel method to analyze the error bounds of the nearest subspace classifier in terms of binary case and applied it

to SRC. Both theoretical and experimental results show the effectiveness of our methods in large scale clustering.

The work may be extended or improved from the following aspects. First, the proposed framework is based on the assumption that out-of-sample data can be represented by in-sample data. Hence, the method may fail to handle the out-of-sample datum when it comes from a new subspaces that does not emerge from in-sample data. It is worth to explore how to overcome this problem in future. Second, the proposed error analysis method only considers the binary case (*i.e.*, $k = 2$). It is more practical but challenging to explore the error analysis method w.r.t. $k > 2$.

ACKNOWLEDGMENT

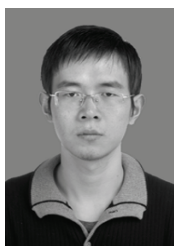
The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper. This work was supported by National Nature Science Foundation of China under grant No.61432012 and No. 61322203.

REFERENCES

- [1] Z. Yu, W. Liu, W. Liu, X. Peng, Z. Hui, and B. V. Kumar, "Generalized transitive distance with minimum spanning random forest," in *Proc. of 24th Int. Joint Conf. on Artif. Intell.*, Buenos Aires, Argentina, Jul. 2015, pp. 2205–2211.
- [2] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, 2001.

- [3] R. Vidal, "Subspace clustering," *IEEE Signal Proc. Mag.*, vol. 28, no. 2, pp. 52–68, 2011.
- [4] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1546–1562, 2007.
- [5] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *Proc. of 21th IEEE Conf. Comput. Vis. and Pattern Recognit.*, Anchorage, AL, Jun. 2008, pp. 1–8.
- [6] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. of 14th Adv. in Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2001, pp. 849–856.
- [7] C. Hou, F. Nie, D. Yi, and D. Tao, "Discriminative embedded clustering: A framework for grouping high-dimensional data," *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1287–1299, Jun. 2015.
- [8] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. of 22th IEEE Conf. Comput. Vis. and Pattern Recognit.*, Miami, FL, Jun. 2009, pp. 2790–2797.
- [9] —, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, 2013.
- [10] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang, "Learning with ℓ^1 -graph for image analysis," *IEEE Trans. on Image Process.*, vol. 19, no. 4, pp. 858–866, 2010.
- [11] D. Xu, Y. Huang, Z. Zeng, and X. Xu, "Human gait recognition using patch distribution feature and locality-constrained group sparse representation," *IEEE Trans. on Image Process.*, vol. 21, no. 1, pp. 316–326, Jan. 2012.
- [12] L. Jing, M. Ng, and T. Zeng, "Dictionary learning-based subspace structure identification in spectral clustering," *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1188–1199, Aug. 2013.
- [13] S. Gao, I.-H. Tsang, and L.-T. Chia, "Laplacian sparse coding, hypergraph laplacian sparse coding, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 92–104, Jan. 2013.
- [14] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. of 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 663–670.
- [15] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, 2013.
- [16] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proc. of 24th IEEE Conf. Comput. Vis. and Pattern Recognit.*, Colorado Springs, CO, Jun. 2011, pp. 1801–1807.
- [17] S. Xiao, M. Tan, and D. Xu, "Weighted block-sparse low rank representation for face clustering in videos," in *Proc. of 13th Eur. Conf. Comput. Vis.*, 2014, pp. 123–138.
- [18] G. C. Liu and S. C. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *Proc. of 13th IEEE Conf. Comput. Vis.*, Barcelona, Spain, Jun. 2011, pp. 1615–1622.
- [19] 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 *Proc. of 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 347–360.
- [20] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *Proc. of 29th AAAI Conf. Artif. Intell.*, Austin, TX, Jan. 2015, pp. 3827–3833.
- [21] R. Liu, Z. Lin, F. D. la Torre, and Z. Su, "Fixed-rank representation for unsupervised visual learning," in *Proc. of 25th IEEE Conf. Comput. Vis. and Pattern Recognit.*, Providence, RI, Jun. 2012, pp. 598–605.
- [22] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *Proc. of 26th IEEE Conf. Comput. Vis. and Pattern Recognit.*, Portland, OR, Jun. 2013, pp. 430–437.
- [23] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009.
- [24] A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Fast l_1 -minimization algorithms and an application in robust face recognition: A review," EECs Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-13, Feb. 2010.
- [25] S. Xiao, W. Li, D. Xu, and D. Tao, "FaLRR: A fast low rank representation solver," in *Proc. of 28th IEEE Conf. Comput. Vis. and Pattern Recognit.*, Boston, MA, Jun. 2015, pp. 4612–4620.
- [26] G. Liu and S. Yan, "Active subspace: Toward scalable low-rank learning," *Neural Comput.*, vol. 24, no. 12, pp. 3371–3394, 2012.
- [27] X. Zhang, F. Sun, G. Liu, and Y. Ma, "Fast low-rank subspace segmentation," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1293–1297, May 2014.
- [28] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, 2004.
- [29] 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, 2011.
- [30] D. Yan, L. Huang, and M. I. Jordan, "Fast approximate spectral clustering," in *Proc. of 15th ACM SIGKDD Int. Conf. Knowl. Dis. and Data Min.*, Paris, France, Jun. 2009, pp. 907–916.
- [31] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proc. of 25th AAAI Conf. Artif. Intell.*, San Francisco, CA, Aug. 2011, pp. 313–318.
- [32] L. Wang, C. Leckie, R. Kotagiri, and J. Bezdek, "Approximate pairwise clustering for large data sets via sampling plus extension," *Pattern Recognit.*, vol. 44, no. 2, pp. 222–235, 2011.
- [33] 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 Trans. Neural. Netw.*, vol. 22, no. 11, pp. 1796–1808, 2011.
- [34] M.-A. Belabbas and P. J. Wolfe, "Spectral methods in machine learning and new strategies for very large datasets," *Proc. of Natl. Acad. Sci.*, vol. 106, no. 2, pp. 369–374, 2009.
- [35] K. Tasdemir, "Vector quantization based approximate spectral clustering of large datasets," *Pattern Recognit.*, vol. 45, no. 8, pp. 3034–3044, 2012.
- [36] 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.
- [37] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. of IEEE Int. Conf. on Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 471–478.
- [38] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, Nov. 2010.
- [39] M. Smith, N. Milic-Frayling, B. Shneiderman, E. Mendes Rodrigues, J. Leskovec, and C. Dunne, "Nodexl: a free and open network overview, discovery and exploration add-in for excel 2007/2010," *Social Media Research Foundation*, 2010.
- [40] U. V. Luxburg, O. Bousquet, and M. Belkin, "Limits of spectral clustering," in *Proc. of 17th Adv. in Neural Inf. Process. Syst.*, Hyatt Regency, Canada, Dec. 2004, pp. 857–864.
- [41] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Sparse representation with kernels," *IEEE Trans. on Image Process.*, vol. 22, no. 2, pp. 423–434, 2013.
- [42] Z. Wang, J. Yang, N. Nasrabadi, and T. Huang, "A max-margin perspective on sparse representation-based classification," in *Proc. of IEEE Conf. Comput. Vis.*, Sydney, Australia, Dec. 2013, pp. 1217–1224.
- [43] J. Hamm and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *Proc. of 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, Jul. 2008, pp. 376–383.
- [44] K. Fukunaga, *Introduction to Statistical Pattern Recogn. (2nd Ed.)*. San Diego, CA: Academic Press Professional, Inc., 1990.
- [45] R. Chitta, R. Jin, T. Havens, and A. Jain, "Approximate kernel k-means: solution to large scale kernel clustering," in *Proc. of 17th ACM SIGKDD Int. Conf. Knowl. Dis. and Data Min.*, San Diego, CA, Aug. 2011, pp. 895–903.
- [46] A. S. Georghiadis, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, 2001.
- [47] 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.
- [48] A. Martinez, "The AR face database," *CVC Technical Report*, vol. 24, 1998.
- [49] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07–49, Oct. 2007.
- [50] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image Vision Comput.*, vol. 28, no. 5, pp. 807–813, 2010.
- [51] M. Yang, L. Zhang, D. Zhang, and S. Wang, "Relaxed collaborative representation for pattern classification," in *Proc. of 25th IEEE Conf. Comput. Vis. and Pattern Recognit.*, Providence, RI, Jun. 2012, pp. 2224–2231.

- [52] D. Cai, X. F. He, and J. W. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data En.*, vol. 17, no. 12, pp. 1624–1637, 2005.
- [53] F. Alimoglu and E. Alpaydin, "Combining multiple representations and classifiers for pen-based handwritten digit recognition," in *Proc. of 4th Int. Conf. Doc. Anal. and Recognit.*, ULM, Germany, Aug. 1997, pp. 637–640.
- [54] J. Blackard and D. Dean, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables," *Comput. Electron. Agric.*, vol. 24, no. 3, pp. 131–151, 1999.
- [55] Y. Taigman, L. Wolf, and T. Hassner, "Multiple one-shots for utilizing class label information," in *Proc. of 20th Brit. Mach. Vis. Conf.*, London, England, Sep. 2009, pp. 1–12.
- [56] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recogn.*, vol. 41, no. 1, pp. 176–190, 2008.
- [57] D. Cai, "Litekmeans: the fastest matlab implementation of kmeans," Available at: <http://www.zjucadcg.cn/dengcai/Data/Clustering.html>, 2011.
- [58] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Mach. Learn.*, vol. 55, no. 3, pp. 311–331, 2004.
- [59] S. Negahban and M. J. Wainwright, "Estimation of (near) low-rank matrices with noise and high-dimensional scaling," *Ann. Stat.*, vol. 39, no. 2, pp. 1069–1097, 2011.
- [60] T. O. Kvalseth, "Entropy and correlation: Some comments," *IEEE Trans. Syst. Man Cybern.*, vol. 17, no. 3, pp. 517–519, 1987.



Xi Peng is a research scientist at Institute for Infocomm., Research Agency for Science, Technology and Research (A*STAR) Singapore. He received the BEng degree in Electronic Engineering and MEng degree in Computer Science from Chongqing University of Posts and Telecommunications, and the Ph.D. degree from Sichuan University, China, respectively. His current research interests include computer vision, image processing, and pattern recognition.

Dr. Peng is the recipient of China National Graduate Scholarship in 2013, CSC-IBM Scholarship for Outstanding Chinese Students in 2012, and Excellent Student Paper of IEEE CHENGDU Section in 2010. He has served as a PC member for 10 international conferences such as IJCNN 2014-2016 and a reviewer for over 10 international journals such as IEEE TNNLS, TIP, TKDE, TIFS, TGRS, TCYB.



Huajin Tang (M'01) received the B.Eng. degree from Zhejiang University, Hangzhou, China, in 1998, the M.Eng. degree from Shanghai Jiao Tong University, Shanghai, China, in 2001, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2005. He was a System Engineer with STMicroelectronics, Singapore, from 2004 to 2006, and then a Post-Doctoral Fellow with the Queensland Brain Institute, University of Queensland, Brisbane, QLD, Australia, from 2006 to 2008. He is currently a

Research Scientist leading the Cognitive Computing Group with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. He has authored one monograph (Springer-Verlag, 2007) and over 30 international journal papers. His current research interests include neural computation, neuromorphic cognitive systems, neurocognitive robots, and machine learning.

Dr. Tang serves as an Associate Editor of the *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS* and an Editorial Board Member of *Frontiers in Robotics and AI*.



Lei Zhang (M'10) received the B.S. and Masters degrees in mathematics and the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2002, 2005, and 2008, respectively. She was a Post-Doctoral Research Fellow in the Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, Hong Kong, from 2008 to 2009. Currently, she is a Professor at Sichuan University, Chengdu. Her current research interests include theory and applications of neural networks

based on neocortex computing and big data analysis methods by infinity deep neural networks.



Zhang Yi (SM'10) received the Ph.D. degree in mathematics from the Institute of Mathematics, The Chinese Academy of Science, Beijing, China, in 1994. Currently, he is a Professor at the College of Computer Science, Sichuan University, Chengdu, China. He is the co-author of three books: *Convergence Analysis of Recurrent Neural Networks* (Kluwer Academic Publisher, 2004), *Neural Networks: Computational Models and Applications* (Springer, 2007), and *Subspace Learning of Neural Networks* (CRC Press, 2010). He is the Chair of

IEEE Chengdu Section (2015). He was an Associate Editor of *IEEE Transactions on Neural Networks and Learning Systems* (2009–2012), and an Associate Editor of *IEEE Transactions on Cybernetics* (2014). His current research interests include Neural Networks and Big Data. He is the founding director of Machine Intelligence Laboratory. He is also the founder of IEEE Computational Intelligence Society, Chengdu Chapter.



Shijie Xiao received the B.E. degree from the Harbin Institute of Technology, Harbin, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Computer Engineering, Nanyang Technological University, Singapore.

His current research interests include machine learning and computer vision.