7-2018

Low-rank sparse subspace for spectral clustering

Xiaofeng ZHU
Guangxi Normal University

Shichao ZHANG
Guangxi Normal University

Yonggang LI
Guangxi Normal University

Jilian ZHANG
Singapore Management University, jilian.z.2007@phdis.smu.edu.sg

Lifeng YANG
Guangxi Normal University

See next page for additional authors

DOI: https://doi.org/10.1109/TKDE.2018.2858782

Author
Xiaofeng ZHU, Shichao ZHANG, Yonggang LI, Jilian ZHANG, Lifeng YANG, and Yue FANG

Published in IEEE Transactions on Knowledge and Data Engineering,
July 2018,
DOI 10.1109/TKDE.2018.2858782

Low-rank Sparse Subspace for Spectral Clustering

Shichao Zhang, *Senior Member, IEEE,* Xiaofeng Zhu, Yonggang Li, Jilian Zhang, Lifeng Yang, and Yue Fang

Abstract—Traditional graph clustering methods consist of two sequential steps, *i.e.,* constructing an affinity matrix from the original data and then performing spectral clustering on the resulting affinity matrix. This two-step strategy achieves optimal solution for each step separately, but cannot guarantee to obtain the globally optimal clustering results. Moreover, the affinity matrix directly learned from the original data will seriously affect the clustering performance, since high-dimensional data are usually noisy and may contain redundancy. To address the above issues, this paper proposes a Low-rank Sparse Subspace (LSS) clustering method via dynamically learning the affinity matrix from low-dimensional space of the original data. Specifically, we learn a transformation matrix to project the original data to their low-dimensional space, by conducting feature selection and subspace learning in the sample self-representation framework. Then we utilize the rank constraint and the affinity matrix directly obtained from the original data to construct a dynamic and intrinsic affinity matrix. Moreover, each of these three matrices is updated iteratively while fixing the other two. In this way, the affinity matrix learned from the low-dimensional space is the final clustering results. Extensive experiments are conducted on both synthetic and real datasets to show that our proposed LSS method outperforms the state-of-the-art clustering methods.

Index Terms—Feature selection, affinity matrix, spectral clustering, subspace learning

1 Introduction

High-dimensional data can be represented by an union of multiple low-dimensional subspaces [1], hence spectral clustering may achieve truthful results by clustering samples according to their underlying subspace. Generally, according to the utilization of graph theory, existing clustering methods can roughly be partitioned into two categories [2], *i.e.,* non-graph clustering methods such as K-means [3], mean-shift [4], Expectation-Maximization (EM) [5] and density based method such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [6], as well as graph clustering methods such as min-cuts [7], Normalized Cuts (NCut) [8], and subspace clustering [9], [10]. Non-graph clustering methods usually achieve the best clustering results when samples are not corrupted by noise and the intrinsic structure of the data is simple. For example, K-means is widely used for data clustering due to simplicity and fair clustering performance, but it performs poorly when the structure of the data is complicated. On the other hand, DBSCAN is difficult to distinguish accurate clusters when the sample space has a cross or the spaces are very close.

To overcome the drawbacks of non-graph clustering methods, graph clustering methods take advantage of correlations among samples to achieve better results by transforming the data partition problem into a graph-cut problem. NCut is a popular graph partitioning method measuring the dissimilarity between two different subspaces and the similarity between samples in the same subspace for learning a similarity or affinity matrix. Another important graph clustering method is spectral clustering, which constructs the affinity matrix by considering various inherent structures of the data such as global structure and local structure [11], [12]. Sparse Subspace Clustering (SSC) method, on the other hand, can also achieve a good clustering result by conducting spectral clustering on the resulting affinity matrix which is a sparse similarity graph of the samples [13]. As a consequence, spectral clustering methods have drawn increasing attention from researchers around the world and have been utilized in many applications.

Usually, spectral clustering consists of two separate steps [14], *i.e.,* contructing an affinity matrix and performing clustering on the generated affinity matrix. Most spectral clustering methods consider the correlation between samples when constructing the affinity matrix. In essence, the affinity matrix can be regarded as a graph, hence the clustering problem is transformed to the problem of computing the optimal graph partitioning. This transformation may remarkably reduce the complexity of clustering and hence play an important role in spectral clustering. To calculate the similarity between two samples, the sample self-representation assumes that each sample can be represented by other samples in the same subspace [12], [15], [16]. Depending on whether a sample can be linearly represented by all the other samples or some of its nearest neighbors, the spectral clustering methods can be divided into two categories,*i.e.,* global representation method

- *This work is partially supported by the China Key Research Program (Grant No: 2016YFB1000905); the Natural Science Foundation of China (Grants No: 61573270 and 61672177); the Project of Guangxi Science and Technology (GuiKeAD17195062); the Guangxi Natural Science Foundation (Grant No: 2015GXNSFCB139011); the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing; the Guangxi Bagui Teams for Innovation and Research; the Guangxi High Institutions Program of Introducing 100 High-Level Overseas Talents; and the Research Fund of Guangxi Key Lab of Multisource Information Mining & Security (18-A-01-01).*

- *S. Zhang, X. Zhu, Y. Li, L. Yang, and Y. Fang are with Guangxi Key Lab of Multi-source Information Mining Security, Guangxi Normal University, Guilin, 541004, China.*
Jilian Zhang is with College of Cyber Security, Jinan University, Guangzhou 510632, China.
Corresponding author: X. Zhu (seanzhuxf@gmail.com).

*Manuscript received April **, 2017; revised ** **, 2017.*

such as LRR [17] and LSR [9], and local representation method such as SSC [13] and CLR [18]. The major difference among previous spectral clustering methods is the learning process of the affinity matrix from original data [9], [17], [19]–[21].

The two-step spectral clustering methods can effectively deal with clustering tasks, however they are likely to produce worse results. The main reason is that the two-step methods may construct a low quality affinity matrix, which is directly learned from the original data. On one hand, the original data are usually high-dimensional, causing the so-called *curse of dimensionality* problem [22]–[24]. On the other hand, the original data contain noise and redundancy as well, and a good affinity matrix cannot be derived from it. As a consequence, these two intrinsic problems of the original data will lead to a low quality affinity matrix, based on which only a suboptimal clustering results can be got [18], [25].

To tackle the above problems, this paper proposes a Low-rank Sparse Subspace (LSS) clustering method, which can dynamically learn an affinity matrix from the intrinsic low-dimensional space of the original data. The proposed LSS utilizes the following procedure to construct an ideal affinity matrix from the samples: 1) perform a sample self-representation process to measure sample similarity, that is, each sample is represented by a subset of all the samples in the same subspace; 2) learn the intrinsic low-dimensional space of the original data by simultaneously conducting subspace learning and feature selection during the representation process; 3) dynamically construct the affinity matrix from the low-dimensional space of the original data; and 4) impose a rank constraint on the Laplacian matrix of the affinity matrix, so that the final clustering results can be derived from the ideal affinity matrix directly. To this end, we integrate three learning processes into a unified framework, *i.e.,* learning the affinity matrix, learning the low-dimensional space, and learning clustering results, so that each of them can be iteratively updated while fixing the other two. As a consequence, the affinity matrix can be learn from the low-dimensional space and the final clustering results can be achieved simultaneously. Such a one-step strategy avoids generating suboptimal results, which is inevitably in the two-step strategy of the existing clustering methods.

We briefly summarize the main contributions of our proposed LSS method as follows:

- Different from the previous graph clustering methods that constructs either a fixed affinity matrix or a dynamic affinity matrix from the original data, LSS learns a dynamic affinity matrix from the intrinsic low-dimensional space of the original data. In addition, LSS constructs affinity matrix and learns the low-dimensional space iteratively, which guarantees a high quality affinity matrix because noise and redundancy are eliminated in the low-dimensional space.
- The low rank constrain is utilized on the Laplacian matrix of the affinity matrix, so as to produce explicitly ideal block structure, in other words, the affinity matrix corresponds to the clustering results. Adaptively adjusting the learning process of the affinity matrix and the learning process of the clustering results, until our LSS method achieves the best clustering results.
- Our LSS method simultaneously integrates three processes into a unified framework, *i.e.,* learning the affinity matrix, learning the low-dimensional space of the original data, and learning clustering results. This is different from the previous clustering methods that consider each process separately.

The paper is organized as follows. In Section 2, we briefly introduce existing clustering methods, and then present our LSS method in Section 3. We evaluate our proposed LSS method through extensive experiments in Section 4. Finally, we conclude the paper in Section 5.

2 Related Work

In this section, we review the most relevant clustering work, *i.e.,* non-graph clustering methods and graph clustering methods.

2.1 Non-graph methods

Non-graph methods performs clustering directly on the original data, where the most representative one is K-means. Given a dataset $\mathbf{D}$, K-means randomly selects K data points from $\mathbf{D}$ as the initial cluster centers, and then alternates the following two steps. In the first step, for each of the data points in $\mathbf{D}$ we compute its distance to each of the K cluster centers, then assign the data point to its nearest cluster. In the second step, we update each of the K cluster centers by considering the member data points in the cluster. K-means will terminate until the assignments no longer change. Obviously, it is difficult to determine a suitable K and when dimensionality of the data is very high, distance computation is very inefficient.

Existing work based on non-graph methods can be roughly divided into three categories: statistics-based, density-based, and hierarchical-based, for instance, BIRCH [26] and CURE [27]. Statistical methods, such as Multi-Stage Learning (MSL) [28] and Mixtures of Probabilistic PCA (MPPCA) [29], assume that the data has a Gaussian distribution inside each subspace and they apply Expectation Maximization (EM) to a mixture of probabilistic PCAs. The main idea of density-based methods is that when the density of a region is larger than a threshold, then the region will be included into the nearby cluster. However, these methods, such as DBSCAN [30] and DENCLUE [31], can only find the quasi-circular clusters. Given a dataset, hierarchical methods directly perform hierarchical decomposition on the data, until some predefined condition is met. This kind of methods is usually distance or density based, and they are easily affected by noise and outliers.

In a nutshell, common limitations of these non-graph methods include: 1) performance is easily affected by the data, and 2) they may encounter the problem of curse of dimensionality.

2.2 Graph based methods

As a useful technique for subspace learning and data clustering, graph-based methods have become popular in machine learning and data mining communities [19], [32]–[34]. Most graph methods firstly construct an affinity matrix to measure similarity between any two data points, and then perform graph cutting or spectral analysis on the resulting affinity matrix [12], [19], [35]. For example, Ncut [8] transfers the data clustering task into a graph partition problem via effectively measuring dissimilarity between different groups and similarity within the same groups, while SSC [13] utilizes a representation-based sparse model to construct an affinity matrix. Low-Rank Representation (LRR) aims to find the low-rank representation of all data and then obtains the clustering results base on their respective subspaces [17]. Least Square Regression (LSR) [9] theoretically illustrates that the affinity matrix is usually presented as a block diagonal structure under an ideal situation, and then obtains the clustering results by making full use of the data correlation. Smooth Representation Clustering [20] takes advantage of grouping effect of the data self-representation model and meanwhile introduces a new grouping effect condition to obtain more effective affinity matrix. GKM [33] takes the intrinsic manifold structure of the data into account to construct an affinity matrix for clustering. Spectral clustering in [34] shows that the dimension of the ambient space is crucial for clustering, based on the assumption that low dimensions chosen in prior work are not optimal. They suggest a lower and a upper bound together with a data-driven procedure for choosing the optimal ambient dimension to construct the affinity matrix. Sparse representation based spectral clustering (SRSC) [36] constructs the affinity matrix by using all of the sparse representation coefficient vectors for spectral clustering. As a supplement of SRSC, NMFSC [37] constructs its affinity matrix by using Nonnegative Matrix Factorization (NMF) coefficient vectors to cluster large scale high-dimensional datasets.

Lots of strategies have been proposed to obtain a desirable affinity matrix. However, the above graph clustering methods employ a two-step strategy, which has been shown to easily result in suboptimal clustering results [25]. To address this issue, Constrained Laplacian Rank (CLR) [18] puts forward a block diagonal affinity matrix and then directly clusters data points into exactly K connected components/clusters. However, the affinity matrix obtained by CLR is learned from the original data, which usually contains noise and redundancy. Therefore, in this paper we focus on graph clustering, aiming to learn an affinity matrix from the low-dimensional space of the original data to derive an optimal clustering result.

3 Approach

3.1 Notations

In this paper, we denote the input sample matrix by $\mathbf{X}$. Moreover, we utilize the normal italic letters, boldface lowercase letters, and boldface uppercase letters to denote scalars, vectors, and matrices, respectively. We summarize symbols used in the paper in Table 1.

TABLE 1
Description of the symbols used in the paper

Symbols	Description
$\mathbf{X}$	the feature matrix of a sample
$\mathbf{x}$	a vector of $\mathbf{X}$
$\mathbf{x}^i$	the i-th row of $\mathbf{X}$
$\mathbf{x}_j$	the j-th column of $\mathbf{X}$
$x_{i,j}$	the element in the i-th row and the j-th column of $\mathbf{X}$
$\|\|\mathbf{X}\|\|_F$	the Frobenius norm of $\mathbf{X}$, *i.e.,* $\|\|\mathbf{X}\|\|_F = \sqrt{\sum_{i,j} \mathbf{x}_{i,j}^2}$
$\|\|\mathbf{X}\|\|_{2,1}$	the $\ell_{2,1}$-norm of $\mathbf{X}$, *i.e.,* $\|\|\mathbf{X}\|\|_{2,1} = \sum_i \sqrt{\sum_j x_{i,j}^2}$
$rank(\mathbf{X})$	the rank of $\mathbf{X}$
$\mathbf{X}^T$	the transpose of $\mathbf{X}$
$tr(\mathbf{X})$	the trace of $\mathbf{X}$
$\mathbf{X}^{-1}$	the inverse of $\mathbf{X}$

3.2 Framework

Assume that $\left\{\mathbf{X}_i | \mathbf{X}_i \in \mathbb{R}^{d\times n_i}\right\}_{i=1}^k$ are the sample sets drawn from k independent subspaces, *i.e.,* each subspace is equivalent to a cluster, where n_i and d denote the number of samples in cluster i and the number of features, respectively. Many existing work have proved that high-dimensional data are usually contributed in some low dimensional subspaces and the samples in the same cluster always belong to the same subspace [13]. Hence, the goal of clustering is to partition the input samples into a number of subspaces such that samples in the same cluster are homogeneous. In addition, samples in the same subspace are more similar whereas samples from different subspaces are less similar even dissimilar. Consequently, the clustering task can be expressed as partitioning the input data matrix $\mathbf{X}$ into k different independence regions (or clusters) according to similarity between samples, where k is the number of clusters.

The clustering performance, however, highly relies on the affinity matrix, and the noise and the redundancy in the original data always result in a low quality affinity matrix. To solve this issue, our proposed LSS method constructs two affinity matrices, *i.e.,* the original affinity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and another intrinsic affinity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$. Specially, the original affinity matrix $\mathbf{A}$ is directly learned from the original data and $\mathbf{S}$ is derived from the low-dimensional space of the original data by conducting feature selection and subspace learning on the transformation matrix $\mathbf{W}$. In this way, LSS can significantly eliminate the impacts of noise and redundancy when constructing an ideal affinity matrix.

Meanwhile, existing two-step based clustering methods obtain the final clustering results by searching the optimal solution at each step. This strategy, however, cannot guarantee that the final clustering results is globally optimal, since a globally optimal clustering solution is not equivalent to combining together the optimal results of the two steps, let along the difficulty to obtain the optimal results at each step. Instead, we impose a rank constraint on the Laplacian matrix of $\mathbf{S}$ to simultaneously achieve an ideal affinity matrix and subsequently the final clustering results. Furthermore, LSS iteratively updates these two affinity matrices and the transformation matrix $\mathbf{W} \in \mathbb{R}^{d\times k}$, by converting the original data into their low-dimensional space according to the rank constraint, until both of them converges. Therefore, $\mathbf{A}$ will approaches to $\mathbf{S}$, meaning that LSS will construct the dynamic and intrinsic affinity matrix $\mathbf{S}$ from the low-dimensional space spanned by $\mathbf{W}^T\mathbf{X}$, according to matrix $\mathbf{A}$ and the low-rank constraint. The flowchart of our LSS method is given in Fig. 1.

3.3 Affinity matrix learning

The affinity matrix is introduced to measure the similarity between any samples. Given that each sample is only connected with its nearest neighbors [1], [11], [38], samples close to each other should have high similarity score while samples far apart should have small or even zero similarity score. Based on the assumption in [39], [40] that the affinity matrix $\mathbf{S}$ that measures similarity among samples can be transformed to guide the prediction of the original feature matrix $\mathbf{X}$, *i.e.,*

$$\min_{\mathbf{W}} = \sum_{i,j=1}^{n} s_{i,j} ||\mathbf{W}^T\mathbf{X}_i - \mathbf{W}^T\mathbf{X}_j||_2^2 \tag{1}$$

where $\mathbf{W}$ is the transformation matrix and $\mathbf{W}^T\mathbf{x}_i$ the prediction of the i-th superpixel $\mathbf{x}_i$. In Eq. (1), the similarity score $s_{i,j}$ between the i-th sample $\mathbf{x}_i$ and the j-th sample $\mathbf{x}_j$ is learned from the original data $\mathbf{X}$ before optimizing the matrix $\mathbf{W}$. Unfortunately, original data usually contains noisy/redundant features, hence producing low quality $\mathbf{S}$. In this case, the assumption that

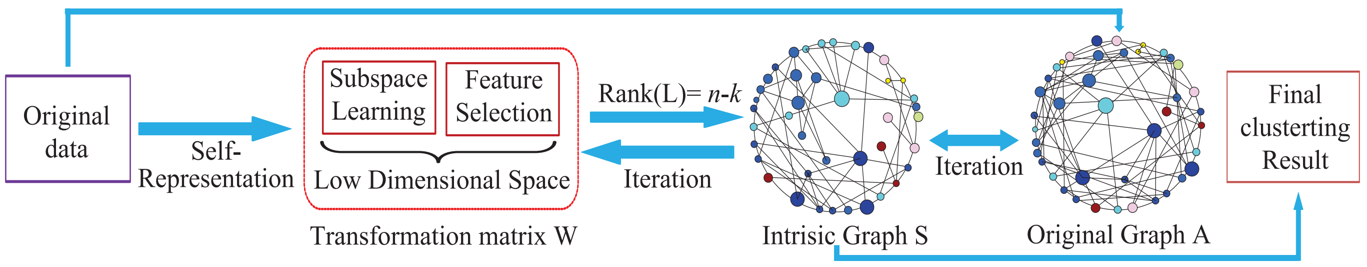


Fig. 1. The flowchart of the proposed LSS method.

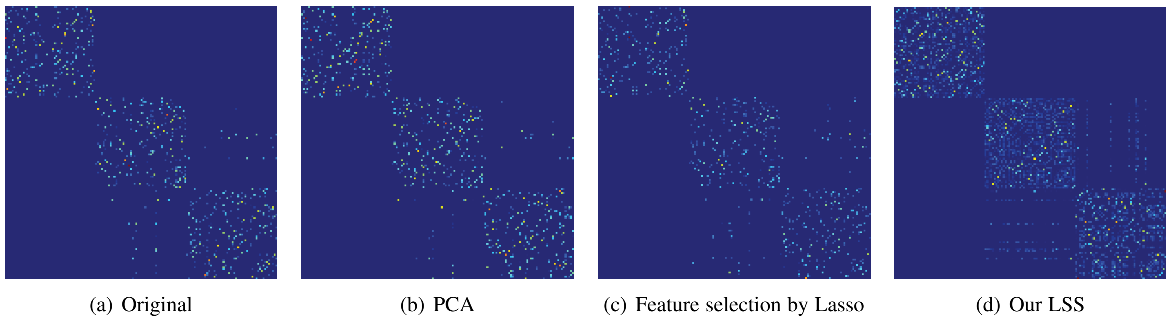


Fig. 2. An illustration of the affinity matrix generated by different feature spaces on dataset iris.

the high-dimensional data has low-dimensional intrinsic space [15], [41] motivates us to search for a high-quality $\mathbf{S}$ from the low-dimensional space spanned by $\mathbf{W}^T\mathbf{X}$. However, the fact is that neither the affinity matrix $\mathbf{S}$ nor the transformation matrix $\mathbf{W}$ are known. To solve this problem, we couple them together in a framework in such a way that they can be iteratively updated, until achieving their optimality respectively. Therefore, we have the following objective function:

$$\begin{aligned}&\min_{\mathbf{S},\mathbf{W}} \|\mathbf{W}^T\mathbf{X} - \mathbf{W}^T\mathbf{X}\mathbf{S}\|_F^2 + \gamma\|\mathbf{W}\|_{2,1},\\ &\quad \forall i, \mathbf{s}_i^T\mathbf{1} = 1, s_{i,i} = 0,\\ &\quad s_{i,j} \geq 0 \text{ if } j \in \mathbb{N}(i), \text{ otherwise } 0\end{aligned} \tag{2}$$

where γ is a parameter to be tuned. The regularization term $\|\mathbf{W}\|_{2,1}$ is used to select features by enforcing matrix $\mathbf{W}$ to contain row sparsity entries, such that noisy/redundant features of $\mathbf{X}$ can be removed. $\|\mathbf{W}^T\mathbf{X} - \mathbf{W}^T\mathbf{X}\mathbf{S}\|_F^2 = \sum_{i,j=1}^n \|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{X}\mathbf{s}_i\|_2^2$ indicates that $\mathbf{x}_i$ is represented by all the samples in $\mathbf{X}$ in their low-dimensional space, *i.e.,* each $\mathbf{W}^T\mathbf{x}_i$ is represented by all the samples $\mathbf{W}^T\mathbf{X}$. The weight matrix $\mathbf{S}$ is the new representation of $\mathbf{X}$, *i.e.,* $\mathbf{S}$ is the affinity matrix of $\mathbf{W^T X}$. The constraint "$\forall i, \mathbf{s}_i^T\mathbf{1} = 1, s_{i,i} = 0,\ s_{i,j} \geq 0 \text{ if } j \in \mathbb{N}(i), \text{ otherwise } 0$" implies that each $\mathbf{W}^T\mathbf{x}_i$ is sparsely represented by all elements in $\mathbf{W}^T\mathbf{X}$, where $j \in \mathbb{N}(i)$ means that the j-th superpixel is one of the nearest neighbors of the i-th superpixel.

Compared with the traditional pairwise similarity measurement in [39], the sparse representation in Eq. (2) has at least the following two advantages. First, the sparse representation in Eq. (2) shows discriminative ability, *i.e.,* Eq. (2) only selects the samples related to the reconstruction of all the samples in the low-dimensional feature space, thus removing the adverse impacts of noisy samples. Second, Eq. (2) iteratively updates $\mathbf{W}$ and $\mathbf{S}$ until both of them converge. During the update, the optimized $\mathbf{S}$ can be used to guide the search for important features, *i.e.,* $\mathbf{W}$. On the other hand, after eliminating the adverse impact of noisy/redundant features, the important features will enable us to find a better sparse representation, *i.e.,* $\mathbf{S}$.

Iteratively updating $\mathbf{S}$ and $\mathbf{W}$ in Eq. (2), however, may generate suboptimal results or lead $\mathbf{S}$ and $\mathbf{W}$ to zero, since the minimization problem in Eq. (2) does not have a constraint to prevent this from happening. As a remedy, we induce another affinity matrix $\mathbf{A}$ that is learned from the original feature space by using the following objective function:

$$\begin{aligned}&\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{X}\mathbf{A}\|_F^2, \quad s.t.,\ \forall i, \mathbf{a}_i^T\mathbf{1} = 1, a_{i,i} = 0,\\ &a_{i,j} \geq 0 \text{ if } j \in \mathbb{N}(i), \text{ otherwise } 0\end{aligned} \tag{3}$$

We build two different affinity matrices $\mathbf{A}$ and $\mathbf{S}$ for data matrix $\mathbf{X}$, according to the observation that different feature spaces result in different clustering results. As shown in Fig. 2, subfigure (a) is the feature space of the original dataset IRIS, whereas subfigures (b)-(d) correspond to feature spaces generated by PCA, Lasso, and our method respectively. Specifically, PCA finds the principal component of the original data and then constructs affinity matrix in the principal component space. For LASSO, we utilize LASSO-based feature selection method to select important features, and then construct the affinity matrix by using the selected features. Different from PCA and LASSO, our LSS uses orthogonal-based low-rank constraint to dynamically choose the real important features, avoiding the obstruction of the redundant features, and finally constructs an affinity matrix with k blocks which exactly corresponds to the final clustering results. It is well-known that the clearer of the affinity matrix, the better the clustering results. Since both $\mathbf{A}$ and $\mathbf{S}$ are different similarity measurement of the same data points, the difference $\sum_{i=1}^n(\|\mathbf{a}_i - \mathbf{s}_i\|_2^2)$ between elements in $\mathbf{A}$ and $\mathbf{S}$ should be as small as possible. Moreover, this difference can be used to guide the iterative optimization process in Eq. (2), so that $\mathbf{A}$ and $\mathbf{S}$ can converge to ideal affinity matrix, respectively. Thus we have the following objective function:

$$\begin{aligned}\min_{\mathbf{A},\mathbf{S},\mathbf{W}} & \|\mathbf{W}^T\mathbf{X}-\mathbf{W}^T\mathbf{X}\mathbf{S}\|_F^2+\alpha\|\mathbf{X}-\mathbf{X}\mathbf{A}\|_F^2\\ & +\beta\textstyle\sum_{i=1}^n\|\mathbf{a}_i-\mathbf{s}_i\|_2^2+\gamma\|\mathbf{W}\|_{2,1},\\ & s.t.\ \forall i, \mathbf{a}_i^T\mathbf{1}=1, a_{i,i}=0,\\ & a_{i,j}\geq 0 \text{ if } j\in\mathbb{N}(i), \text{ otherwise } 0;\\ & \forall i, \mathbf{s}_i^T\mathbf{1}=1, s_{i,i}=0,\\ & s_{i,j}\geq 0 \text{ if } j\in\mathbb{N}(i), \text{ otherwise } 0.\end{aligned}\tag{4}$$

where α, β and γ are parameters to tune, and constraint $\sum_{i=1}^n\|\mathbf{a}_i-\mathbf{s}_i\|_2^2$ is used to preserve consistency of $\mathbf{A}$ and $\mathbf{S}$.

3.4 Low-rank sparse subspace clustering

The optimization model in Eq. (4) can produce high quality affinity matrix $\mathbf{S}$ (or $\mathbf{A}$), but it still involves a graph-cut problem that is NP-hard, failing to explicitly generate the segmentation results. To overcome this drawback, we introduce the final objective function of our LSS method as follows:

$$\begin{aligned}\min_{\mathbf{A},\mathbf{S},\mathbf{W}} & \|\mathbf{W}^T\mathbf{X}-\mathbf{W}^T\mathbf{X}\mathbf{S}\|_F^2+\alpha\|\mathbf{X}-\mathbf{X}\mathbf{A}\|_F^2\\ & +\beta\textstyle\sum_{i=1}^n\|\mathbf{a}_i-\mathbf{s}_i\|_2^2+\gamma\|\mathbf{W}\|_{2,1},\\ & s.t.\ \forall i, \mathbf{a}_i^T\mathbf{1}=1, a_{i,i}=0,\\ & a_{i,j}\geq 0 \text{ if } j\in\mathbb{N}(i), \text{ otherwise } 0;\\ & \forall i, \mathbf{s}_i^T\mathbf{1}=1, s_{i,i}=0,\\ & s_{i,j}\geq 0 \text{ if } j\in\mathbb{N}(i), \text{ otherwise } 0;\\ & \mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}=\mathbf{I}_k,\end{aligned}\tag{5}$$

where $\mathbf{W}\in\mathbb{R}^{d\times k}$ and $\mathbf{I}_k\in\mathbb{R}^{k\times k}$ are the transformation matrix and the identity matrix, respectively.

There is only one difference between Eq. (4) and Eq. (5), that is, Eq. (5) has one more constraint $\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}=\mathbf{I}_k$. In Theorem 1 we prove that this extra constraint enables Eq. (5) to generate explicit clustering solution, *i.e.,* resultant matrix $\mathbf{S}$ will have exact k connected components (or blocks).

Theorem 1. *The optimal* $\mathbf{S}$ *generated by Eq. (5) has* k *blocks (or connected components), where* k *is the number of clusters.*

Proof. First, by following the Ky Fan's theorem in [42], we have the following Lemma:

Lemma 1.

$$\left\{\begin{array}{l}\min\limits_{\mathbf{W}}\ tr(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}),\\ s.t., \mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}=\mathbf{I}_k.\end{array}\right. \Leftrightarrow \textstyle\sum_{i=1}^k\lambda_i\rightarrow 0\tag{6}$$

where $\mathbf{I}_n$ is an $n\times n$ identity matrix, $\mathbf{L}=(\mathbf{I}_n-\mathbf{S})(\mathbf{I}_n-\mathbf{S})^T$, and $\lambda_i, i\in 1,...,k$ is the least k eigenvalues of $\mathbf{L}$.

Second, $\sum_{i=1}^k\lambda_i\rightarrow 0$ can be regarded as the relaxation version of the constraint "$\mathbf{L}$ has k zero eigenvalues", *i.e.,* $rank(\mathbf{L})=n-k$.

Third, by following [43], [44], we have the following Lemma:

Lemma 2. *The number of eigenvalue 0 of the Laplacian matrix* $\mathbf{L}$, i.e., $rank(\mathbf{L})=n-k$, *is equal to the number of the connected components of the affinity matrix* $\mathbf{S}$.

Finally, we know the constraint $\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}=\mathbf{I}_k$ ensures that $\mathbf{S}$ has exactly k connected components (or blocks), so $\mathbf{S}$ has explicit clustering results. Due to this constraint is related to the rank of the Laplacian matrix of the affinity matrix, so we call it the rank constraint in this paper.

□

In Eq. (5), we integrate the learning of two affinity matrices $\mathbf{A}$ and $\mathbf{S}$, the transformation matrix $\mathbf{W}$, and the low-rank constraint $\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}=\mathbf{I}_k$, into a unified framework. In the framework, a robust feature selection model based on $\mathbf{W}$ can be constructed through 1) the guidance from both the low-rank constraint and the two optimized affinity matrices, and 2) conducting subspace learning (via the row-rank constraint) and feature selection (via the $\ell_{2,1}$-norm) simultaneously. The resulting matrix $\mathbf{W}$ can better help the optimization of $\mathbf{S}$. Meanwhile, the constraint $\sum_{i=1}^n\|\mathbf{a}_i-\mathbf{s}_i\|_2^2$ enforces a tradeoff between the original affinity matrix $\mathbf{A}$ and the affinity matrix $\mathbf{S}$ that is learned from the low-dimensional space. Hence, the low-rank constraint suggests that $\mathbf{S}$ is the final clustering results, and both $\mathbf{A}$ and $\mathbf{S}$ converge to ideal affinity matrix respectively.

Algorithm 1 Framework of the optimization of Eq. (5)

```
Require: $\mathbf{X}\in\mathbb{R}^{n\times d}$, k, $\alpha$, $\beta$ and $\gamma$.
Ensure: Initial $\mathbf{W}$, $\mathbf{A}$, $\mathbf{S}$ randomly.
1: repeat
2:   Obtain $\mathbf{W}$ by solving Eq. (7);
3:   Obtain $\mathbf{A}$ by solving Eq. (9);
4:   Obtain $\mathbf{S}$ by solving Eq. (13);
5: until $rank(\mathbf{L})=n-k$;
Ensure: Optimal affinity matrix $\mathbf{S}$.
```

3.5 Optimization

It is clear that Eq. (5) is not jointly convex on $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{S}$, but is convex on each variable while fixing the rest. In this paper, we employ the alternative optimization strategy to optimize Eq. (5), *i.e.,* iteratively optimizing each variable while fixing the rest until the algorithm converges. The pseudo-code of the framework of our method is given in Algorithm 1.

3.5.1 Update $\mathbf{W}$ *while fixing* $\mathbf{S}$ *and* $\mathbf{A}$

When $\mathbf{S}$ and $\mathbf{A}$ are fixed, Eq. (5) turns into the following optimal problem:

$$\min_{\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}=\mathbf{I}_c}\|\mathbf{W}^T\mathbf{X}-\mathbf{W}^T\mathbf{X}\mathbf{S}\|_F^2+\gamma\|\mathbf{W}\|_{2,1},\tag{7}$$

The objective function in Eq. (7) is convex with respect to $\mathbf{W}$, but non-smooth due to the term $\|\mathbf{W}\|_{2,1}$. In this paper, we employ the framework of iteratively reweighted least square in [45] to optimize $\mathbf{W}$, by iteratively optimizing $\mathbf{W}$ and $\mathbf{Q}$ until converged. Here, $\mathbf{Q}$ is a diagonal matrix with the i-th diagonal element $q_{i,i}=\frac{1}{2||\mathbf{w}^i||_2^2}$. Thus, Eq. (7) is changed to Eq. (8):

$$\min_{\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}=\mathbf{I}_c}\|\mathbf{W}^T\mathbf{X}-\mathbf{W}^T\mathbf{X}\mathbf{S}\|_F^2+\gamma tr(\mathbf{W}^T\mathbf{Q}\mathbf{W}),\tag{8}$$

Eq.(8) is an orthogonal optimization problem and we can solve it by using technique in [46]. We list the pseudo-code for optimizing $\mathbf{W}$ in Algorithm 3, where $\nabla F=\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}+\gamma\mathbf{Q}\mathbf{W}$ is the derivative of Eq. (8). Meanwhile, since $\mathbf{Q}$ dependents on $\mathbf{W}$, we iteratively optimize it by using Algorithm 2:

3.5.2 Update $\mathbf{A}$ *while fixing* $\mathbf{W}$ *and* $\mathbf{S}$

By fixing $\mathbf{W}$ and $\mathbf{S}$, we have the following objective function:

$$\begin{aligned}\min_{\mathbf{A}}\ & \alpha\|\mathbf{X}-\mathbf{X}\mathbf{A}\|_F^2+\beta\textstyle\sum_{i=1}^n\|\mathbf{a}_i-\mathbf{s}_i\|_2^2,\\ & s.t.,\forall i, \mathbf{a}_i^T\mathbf{1}=1, a_{i,i}=0, a_{i,j}\geq 0\end{aligned}\tag{9}$$

Algorithm 2 Framework for solving Eq. (7)

```
Require: $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{S} \in \mathbb{R}^{n\times n}$, $\mathbf{W} \in \mathbb{R}^{d\times c}$ and $\gamma$.
Ensure: $\mathbf{W}$
1: repeat
2:   Obtain $\mathbf{W}^{(t+1)}$ by using Algorithm 3 ;
3:   Calculate the diagonal matrix $\mathbf{Q}^{t+1}$, where the $i$-th diagonal element of $\mathbf{Q}_{t+1}$ is $\frac{1}{2\|\mathbf{w}^{(t+1)^i}\|_2^2}$;
4: until convergence;
```

Algorithm 3 Framework for solving Eq. (8)

```
Require: $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{S} \in \mathbb{R}^{n\times n}$ and $\gamma$.
Ensure: $\mathbf{W}$
1: Initial $\mathbf{W}^{(0)}$;
2: $t \leftarrow 0$;
3: repeat
4:   $\mathbf{W} \leftarrow \mathbf{W}^{(t)}\nabla\mathbf{F}^{T^{(t)}} - \nabla\mathbf{F}^{(t)}\mathbf{W}^{T^{(t)}}$;
5:   $\tau \leftarrow$ non-monotonic line search;
6:   $\mathbf{W}^{t+1} \leftarrow (\mathbf{I} - \frac{\tau}{2}\mathbf{H})^{-1}(\mathbf{I} + \frac{\tau}{2}\mathbf{H})\mathbf{X}^{(t)}$;
7:   $t \leftarrow t+1$;
8: until convergence;
```

The Lagrangian formulation of Eq. (9) can be changed to

$$\min_{\mathbf{A}} \|\mathbf{R} - \mathbf{B}\mathbf{A}\|_{\mathbf{F}}^{\mathbf{2}} \quad s.t.,\ a_{i,i} = 0, \tag{10}$$

where $\mathbf{R} = [\mathbf{X}^T, \sqrt{\beta/\alpha}\mathbf{S}^T, \xi\mathbf{1}]^T$ and $\mathbf{B} = [\mathbf{X}^T, \sqrt{\beta/\alpha}\mathbf{I}^T, \xi\mathbf{1}]^T$, $\mathbf{I} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ is an identify matrix, $\mathbf{1} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ is a square matrix with all-one-element, and ξ approaches to infinity. In this paper, we utilize the alternative optimization method to solve the problem in Eq. (10). Since the optimization of $\mathbf{a}_i$ $(i = 1, ..., n)$ is independent on the optimization of the other $\mathbf{a}_j(j \neq i)$, we optimize $\mathbf{a}_i$ while fixing the other $\mathbf{a}_j$s by using the following objective function:

$$\min_{\mathbf{a_i}} \|\mathbf{R_1} - \mathbf{b}_i\mathbf{a}_i^T\|_2^2 \quad s.t.,\ a_{i,i} = 0, \tag{11}$$

where $\mathbf{R_1} = \mathbf{R} - (\mathbf{B}\mathbf{A} - \mathbf{b}_i\mathbf{a}_i^T)$. Eq. (11) can further be changed to Eq. (12) as follows:

$$\min_{\mathbf{a}_i} \|\mathbf{a}_i - \mathbf{v}_i\|_2^2 \quad s.t.,\ a_{i,i} = 0, \tag{12}$$

where $\mathbf{v}_i = \mathbf{R}_1^T\mathbf{b}_i/\mathbf{b}_i^T\mathbf{b}_i$. Eq. (12) has a closed form solution, *i.e.*, $a_{i,j} = v_{i,j}, j \neq i$ and $a_{i,i} = 0$, where $a_{i,j}$ and $v_{i,j}$ are the j-th element of $\mathbf{a}_i$ and $\mathbf{v}_i$, respectively.

Since the constraint "$\forall i, \mathbf{s}_i^T\mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0$" in Eq. (5) outputs sparse representation for all the data points [47], we follow [18] to set the maximum number of neighbors to k to fix variable $\beta = \frac{1}{n}\sum_{i=1}^n(\frac{k}{2}\hat{v}_{i,k+1} - \frac{1}{2}\sum_{f=1}^k \hat{v}_{i,f})$, where $\hat{\mathbf{v}}_i$ is $\mathbf{v}_i$, $i = 1, ..., n$ on descend order.

3.5.3 *Update* $\mathbf{S}$ *while fixing* $\mathbf{W}$ *and* $\mathbf{A}$

When fixing $\mathbf{W}$ and $\mathbf{A}$, Eq. (5) can be changed to:

$$\begin{aligned}\min_{\mathbf{S}} \|\mathbf{W}^T\mathbf{X} - \mathbf{W}^T\mathbf{X}\mathbf{S}\|_F^2 + \alpha\|\mathbf{A} - \mathbf{S}\|_F^2,\\ s.t., \forall i, \mathbf{s}_i^T\mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0\end{aligned} \tag{13}$$

The Lagrangian function of Eq. (13) can be changed to:

$$\min_{\mathbf{S}} \|\tilde{\mathbf{R}} - \tilde{\mathbf{B}}\mathbf{S}\|_{\mathbf{F}}^{\mathbf{2}} \quad s.t.,\ s_{i,i} = 0 \tag{14}$$

where $\tilde{\mathbf{R}} = [\mathbf{W}\mathbf{X}^{\mathbf{T}}, \sqrt{\mathbf{1}/\alpha}\mathbf{A}^{\mathbf{T}}, \phi\mathbf{1}]$ and $\tilde{\mathbf{B}} = [\mathbf{W}\mathbf{X}^{\mathbf{T}}, \sqrt{\mathbf{1}/\alpha}\mathbf{I}, \phi\mathbf{1}]$, and ϕ approaches to infinity.

Since optimizing $\mathbf{s}_i$ $(i = 1, ..., n)$ is independent on optimizing $\mathbf{s}_j, j \neq i$, we optimize $\mathbf{s}_i$ while fixing $\mathbf{s}_j, j \neq i$ by using the following objective function:

$$\min_{\mathbf{s}_i} \|\tilde{\mathbf{R}}_\mathbf{1} - \tilde{\mathbf{b}}_i\mathbf{s}_i^T\|_2^2 \quad s.t.,\ s_{i,i} = 0, \tag{15}$$

where $\tilde{\mathbf{R}}_\mathbf{1} = \mathbf{R} - (\tilde{\mathbf{B}}\mathbf{S} - \tilde{\mathbf{b}}_i\mathbf{s}_i^T)$. In fact, Eq. (15) can be further changed to Eq. (16) below:

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i - \tilde{\mathbf{v}}_i\|_2^2 \quad s.t.,\ s_{i,i} = 0, \tag{16}$$

where $\tilde{\mathbf{v}}_i = \tilde{\mathbf{R}}_1^T\tilde{\mathbf{b}}_i/\tilde{\mathbf{b}}_i^T\tilde{\mathbf{b}}_i$. And Eq. (16) has a closed form solution, *i.e.*, $s_{i,j} = \tilde{v}_{i,j}, j \neq i$ and $s_{i,i} = 0$. Here, $s_{i,j}$ and $\tilde{v}_{i,j}$ are the j-th element of $\mathbf{s}_i$ and $\tilde{\mathbf{v}}_i$, respectively.

3.6 Convergence analysis

In this section, we analyze and prove the convergence of our proposed Algorithm 1-3 when solving the final objective function Eq. (5). Algorithm 2 and Algorithm 3 have been proved to converge by [46] and [45], respectively, so we prove the convergence of Algorithm 1 by using Theorem 1 given below.

Theorem 1. *Algorithm 1 monotonically decreases the objective function value of Eq. (5) until converges.*

Proof. Assume after the t-th iteration we have obtained the optimal $\mathbf{A}^{(t)}$, $\mathbf{W}^{(t)}$ and $\mathbf{S}^{(t)}$. In the $(t+1)$-th iteration, we need to optimize $\mathbf{S}^{(t+1)}$ while fixing $\mathbf{A}^{(t)}$ and $\mathbf{W}^{(t)}$, as shown below.

3.6.1 *Update* $\mathbf{S}^{(\mathbf{t+1})}$ *while fixing* $\mathbf{A}^{(\mathbf{t})}$ *and* $\mathbf{W}^{(\mathbf{t})}$

.

According to Section 3.5.3, $s_{i,j}^{(t+1)}$ has a closed-form optimal solution, *i.e.*, a global solution, for all $i, j = 1, ..., n$. Thus we have the following inequality:

$$\begin{aligned}&\left\|\mathbf{W}^{(t)T}\mathbf{X} - \mathbf{W}^{(t)T}\mathbf{X}\mathbf{S}^{(t+1)}\right\|_F^2 + \alpha\|\mathbf{X} - \mathbf{X}\mathbf{A}^{(t)}\|_F^2\\ &+\beta\sum_{i=1}^n\left\|\mathbf{a}_i^{(t)} - \mathbf{s}_i^{(t+1)}\right\|_2^2 + \gamma\left\|\mathbf{W}^{(t)}\right\|_{2,1}\\ &\leq \left\|\mathbf{W}^{(t)T}\mathbf{X} - \mathbf{W}^{(t)T}\mathbf{X}\mathbf{S}^{(t)}\right\|_F^2 + \alpha\|\mathbf{X} - \mathbf{X}\mathbf{A}^{(t)}\|_F^2\\ &+\beta\sum_{i=1}^n\left\|\mathbf{a}_i^{(t)} - \mathbf{s}_i^{(t)}\right\|_2^2 + \gamma\left\|\mathbf{W}^{(t)}\right\|_{2,1}\end{aligned} \tag{17}$$

3.6.2 *Update* $\mathbf{A}^{(\mathbf{t+1})}$ *while fixing* $\mathbf{S}^{(\mathbf{t+1})}$ *and* $\mathbf{W}^{(\mathbf{t})}$

.

According to Section 3.5.2, $a_{i,j}^{(t+1)}$ has a closed-form optimal solution, *i.e.*, a global solution, for all $i, j = 1, ..., n$, which implies the following:

$$\begin{aligned}&\left\|\mathbf{W}^{(t)T}\mathbf{X} - \mathbf{W}^{(t)T}\mathbf{X}\mathbf{S}^{(t+1)}\right\|_F^2 + \alpha\|\mathbf{X} - \mathbf{X}\mathbf{A}^{(t+1)}\|_F^2\\ &+\beta\sum_{i=1}^n\left\|\mathbf{a}_i^{(t+1)} - \mathbf{s}_i^{(t+1)}\right\|_2^2 + \gamma\left\|\mathbf{W}^{(t)}\right\|_{2,1}\\ &\leq \left\|\mathbf{W}^{(t)T}\mathbf{X} - \mathbf{W}^{(t)T}\mathbf{X}\mathbf{S}^{(t+1)}\right\|_F^2 \alpha\|\mathbf{X} - \mathbf{X}\mathbf{A}^{(t)}\|_F^2\\ &+\beta\sum_{i=1}^n\left\|\mathbf{a}_i^{(t)} - \mathbf{s}_i^{(t+1)}\right\|_2^2 + \gamma\left\|\mathbf{W}^{(t)}\right\|_{2,1}\end{aligned} \tag{18}$$

3.6.3 Update $\mathbf{W}^{(t+1)}$ while $\mathbf{S}^{(t+1)}$ and $\mathbf{A}^{(t+1)}$

.

According to the conclusion in [45], we can easily have the following:

$$\begin{aligned}&\left\|\mathbf{W}^{(t+1)T}\mathbf{X}-\mathbf{W}^{(t+1)T}\mathbf{X}\mathbf{S}^{(t+1)}\right\|_F^2+\gamma\left\|\mathbf{W}^{(t+1)}\right\|_{2,1}\\&+\alpha\|\mathbf{X}-\mathbf{X}\mathbf{A}^{(t+1)}\|_F^2+\beta\sum_{i=1}^n\left\|\mathbf{a}_i^{(t+1)}-\mathbf{s}_i^{(t+1)}\right\|_2^2\\&\leq\left\|\mathbf{W}^{(t)T}\mathbf{X}-\mathbf{W}^{(t)T}\mathbf{X}\mathbf{S}^{(t+1)}\right\|_F^2+\gamma\left\|\mathbf{W}^{(t)}\right\|_{2,1}\\&+\alpha\|\mathbf{X}-\mathbf{X}\mathbf{A}^{(t+1)}\|_F^2+\beta\sum_{i=1}^n\left\|\mathbf{a}_i^{(t+1)}-\mathbf{s}_i^{(t+1)}\right\|_2^2\end{aligned}\tag{19}$$

By integrating Eq. (17) with Eq. (18) and Eq. (19) , we obtain:

$$\begin{aligned}&\left\|\mathbf{W}^{(t+1)T}\mathbf{X}-\mathbf{W}^{(t+1)T}\mathbf{X}\mathbf{S}^{(t+1)}\right\|_F^2+\gamma\left\|\mathbf{W}^{(t+1)}\right\|_{2,1}\\&+\alpha\|\mathbf{X}-\mathbf{X}\mathbf{A}^{(t+1)}\|_F^2+\beta\sum_{i=1}^n\left\|\mathbf{a}_i^{(t+1)}-\mathbf{s}_i^{(t+1)}\right\|_2^2\\&\leq\left\|\mathbf{W}^{(t)T}\mathbf{X}-\mathbf{W}^{(t)T}\mathbf{X}\mathbf{S}^{(t)}\right\|_F^2+\gamma\left\|\mathbf{W}^{(t)}\right\|_{2,1}\\&+\alpha\|\mathbf{X}-\mathbf{X}\mathbf{A}^{(t)}\|_F^2+\beta\sum_{i=1}^n\left\|\mathbf{a}_i^{(t)}-\mathbf{s}_i^{(t)}\right\|_2^2\end{aligned}\tag{20}$$

Eq. (20) indicates that the objective function value of Eq. (5) decreases after each iteration of Algorithm 1. This concludes the proof of Theorem 1. ☐

3.7 Complexity analysis

The computational complexity of our LSS method mainly depends on the iterative optimization of $\mathbf{W}$, $\mathbf{A}$ and $\mathbf{S}$ in Algorithm 1. Specifically, the time complexity of the optimization of $\mathbf{W}$ is $O(2dn^2+4nd^2+2kd^2+k^3)$, where d and n are the number of features and the number of training samples, respectively [46]. Since the optimization of both $\mathbf{A}$ and $\mathbf{S}$ takes closed form solutions, it has a complexity of $O(n^2)$. Therefore, the overall time complexity of Algorithm 1 is $O(2dn^2+4nd^2+2kd^2+k^3)$. For comparison, time complexity of most clustering methods, such as Constrained Laplacian Rank (CLR) [18], Ratio Cut (RCut) [48], Sparse Subspace Clustering (SSC) [13], Low-Rank Representation (LRR) [17], Smooth Representation (SMR) [20], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [6] and LeastSquares Regression (LSR) [9]), are quadratic with respect to the sample size.

4 Experiments

In this section, we evaluate the performance of our Low-rank Sparse Subspace (LSS) method, by comparing LSS with nine state-of-the-art clustering methods on two synthetic datasets and eight real datasets. We adopt ACC and NMI as performance measures for the algorithms.

4.1 Experiment setting

4.1.1 Dataset

We generate two synthetic datasets, as shown in Fig. 3, to verify the effectiveness of our LSS method. Synthetic dataset 1 (see the left-most column the first row in Fig. 3) includes 400 data points that can be divided into two clusters. Each cluster has 200 data points and some samples in different cluster are close in distance. The purpose of using this kind of synthetic data is to evaluate the ability of clustering methods to correctly recognize the two clusters, without being misled by those neighboring samples from different clusters. We denote Synthetic data 1 as Syn1. Synthetic dataset 2 (see the left-most column of the second row in Fig. 3) contains two clusters that form a circle and a upper case letter '**T**' in the circle. We create this synthetic dataset to verify the ability of clustering methods to separate these two clusters correctly. Obviously, clustering task on Synthetic dataset 2 is more challenging than that on Synthetic dataset 1. We denote Synthetic dataset 2 as Syn2.

TABLE 2
Characteristics of eight benchmark real datasets.

Datasets	Samples	Dimensions	Classes
Coil20	1440	1024	20
Ecoli	336	344	8
Umist	575	644	20
Glass	214	9	6
Usps	1000	256	10
Jaffe	213	1024	10
Tox	171	5748	4
Yeast	1484	1470	10

We also use eight real datasets in our experiments. Specifically, dataset Coil [49] includes 1440 grid images of 20 objects and each image has 1024 features. It is worth noting that the background of all the images in Coil has been discarded. Umist [50] consists of 575 face images of 20 people, where each image contains 23 $\times$ 28 pixels and all images have different poses from profile to frontal views. Ecoli and Glass are widely used for clustering tasks and can be downloaded from UCI Machine Learning Repository [1]. Usps [51] contains 1000 digital images where each of which is a number from 0 to 9, thus Usps can be segmented into 10 groups. Jaffe [52] consists of 213 face images of 10 women and these images have different facial expressions. TOX [53] is a dataset that is followed by observations of high-fat food feeding in mice with 24 weeks. It is used for providing insight into the effect of high fat diets on metabolism in the liver. And Yeast [54] contain 1484 samples with 1470 features for predicting the Cellular localization sites of proteins. Detail statistics of the above real datasets are summarized in Table 2.

4.1.2 Comparison methods

In order to verify (1) how robust our proposed method is with respect to noise and redundancy in data, and (2) how well our one-step method performs compared to existing state-of-the-art two-step clustering methods on real datasets, we use the following methods for comparison: (1) three classic clustering methods, *i.e.,* k-means [55], Ratio Cut (RCut) [48], and Normalized cut (NCut) [8], (2) a density-based clustering method DBSCAN [6], (3) four baseline graph-cut-based methods, *i.e.,* Sparse Subspace Clustering (SSC) [13], Low-Rank Representation (LRR) [17], Smooth Representation (SMR) [20], and Least Squares Regression (LSR) [9], (4) and state-of-the-art method, *i.e.,* Constrained Laplacian Rank (CLR) [18]). We summarize each of these comparison methods as follows.

- The classic k-means clustering method [55] is designed to partition the dataset into groups so that data points in the same groups are homogenous and data points between different groups are heterogeneous.

1. http://archive.ics.uci.edu/ml/.

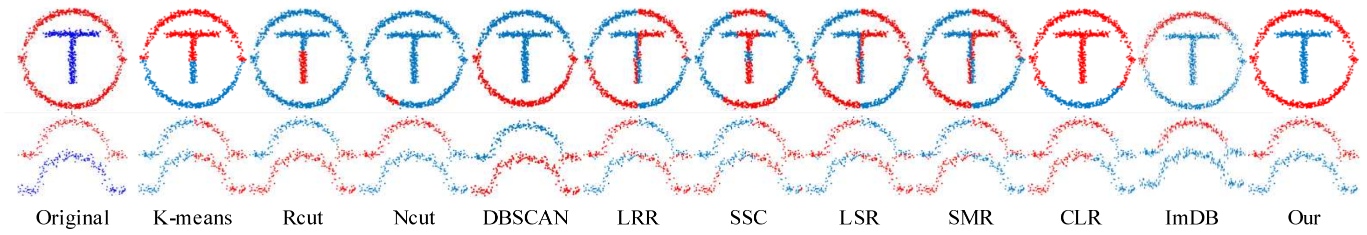


Fig. 3. Experiment results of all the methods on synthetic datasets Syn1 (Top row: the 'circle-and-T' shape) and Syn2 (Bottom row: the 'double-hump'-T' shape).

- Ratio Cut (RCut) [48] captures both the minimum-cut and minimum-width bisection naturally, which are the two traditional goals of segmentation.
- Normalized cuts (NCut) [8] first extracts the global representation of images and then performs clustering task by regarding it as a graph partitioning problem. This classic graph segmentation method uses a two-step strategy.
- Sparse Subspace Clustering (SSC) [13] first learns a sparse similarity matrix using an ℓ_1-norm regularizer, so as to find the intrinsic low-dimensional subspaces of the original high-dimensional space. Then it obtains the final clustering results by conducting spectral clustering on the sparse similarity matrix.
- Low-Rank Representation (LRR) [17] first searches for the low-rank representation of the data via an $\ell_{2,1}$-norm regularizer, and then constructs a undirected graph using the Ncut method.
- Smooth Representation (SMR) method [20] makes use of the least square loss function and the trace norm regularizer to enforce grouping effect among representation coefficients, and then performs subspace segmentation.
- Least Squares Regression (LSR) [9] first takes advantage of correlation of data points, and then obtains clustering results by performing spectral clustering on the learned affinity matrix.
- Constrained Laplacian Rank (CLR) [18] imposes the rank constraint on the Laplacian graph of the affinity matrix, which guarantees that the sparse matrix contains exact k connected components.
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [6] defines a cluster as the maximum set of points connected to density. It groups a region with sufficiently dense data points inside as a cluster.
- Improved DBSCAN Algorithm for detecting stops in Individual Trajectories (ImDB) [56] is a variant of DBSCAN algorithm. ImDB is a hybrid feature-based, density measurement method that takes temporal and spatial properties into account.

For the above competitor methods, NCut, SSC, LRR, SMR and LSR are two-step strategy-based, while DBSCAN is density-based. Although our LSS and CLR are both one-step based method, there are two major differences between them. First, CLR only learns one affinity matrix $\mathbf{S}$, while our LSS learns both $\mathbf{S}$ and $\mathbf{A}$. Second, CLR obtains $\mathbf{S}$ from the original feature space, while LSS learns $\mathbf{S}$ from the low-dimensional space by conducting subspace learning and feature selection simultaneously and $\mathbf{A}$ which could be seen as a guide from original space.

4.1.3 *Experiment set-up*

In our experiments, firstly, we test the robustness of our proposed method, compared with the comparison methods, at both the synthetic datasets containing pre-defined noise and redundancy and the real datasets containing uncertain noise and redundancy, in terms of two evaluation metrics widely used for the study of clustering tasks. Secondly, we investigate the parameters' sensitivity of our proposed method (*i.e.,* α and γ in Eq. (5) via varying their values to observe the variation of clustering performance. Thirdly, we demonstrate the convergence of our proposed algorithm (*i.e.,* Algorithm 1) to solve our proposed objective function Eq. (5) via checking the iteration times while Algorithm 1 converges, *i.e.,* the objective function value in Eq. (5) is stable.

4.2 Performance on synthetic datasets

Fig. 3 shows the clustering results of our LSS and the nine comparison methods on the two synthetic datasets Syn1 (the 'circle-and-T' shape) and Syn2(the 'double-hump' shape).

For Syn1 that contains some neighboring samples from different classes, LSS can perfectly separate the two groups of samples, whereas all the night comparison methods fail to do that. It is worth noting that among the nine comparison methods, Rcut, Ncut perform better than the rest, because they only fail to distinguish a small proportion of neighboring samples. Similar observation of DBSCAN can be found on Syn1, where some samples are assigned the wrong class label.

Performing clustering task on Syn2 is more challenging than on Syn1, since in Syn2 one cluster is completely surrounded by the other cluster. Clearly, all the nine comparison methods are failed to correctly separate the two clusters. And Ncut performs the worse, since it nearly assigns all samples in one cluster, except for those samples in red from the circle-shape cluster. From the results on Syn2, we can see that DBSCAN horizontally separates the samples in the circle-shaped class into the upper part (in blue) and the lower part (in red). And the samples in the T-shaped class are wrongly assigned to the upper part. In contrast, our LSS outperforms the nine comparison methods, correctly isolating the $\mathbf{T}$-shape cluster from the circle-shape cluster.

4.3 Performance on real datastes

We also evaluate our LSS and the nine comparison methods on eight real datasets, with respect to the widely used evaluation criteria, standard clustering accuracy (ACC) and Normalized Mutual Information (NMI). Here, ACC is defined as the total number of correctly classified samples, with respect to their ground truth labels. Obviously, the larger number of corrected classified samples, the higher the ACC value. NMI measures the matching rate between the ground truth labels and the predicted labels. For fairness, in the experiment we set parameter k of K-means method to the true number of clusters, and set the parameters of all the methods in the range [10^{-3}, ..., 10^{3}]. The experiment results are given in Table 4.4.

From Table 4.4 we can see that in terms of ACC, our LSS performs the best on nearly all the eight datasets except for Umist, on which the CLR method only slightly better than LSS. On the other hand, when considering NMI values LSS also outperforms the nine comparison methods on all the real datasets except for Coil20 and Umist, upon which DBSCAN and CLR are slightly better than our LSS method, respectively.

For example, the ACC results of our LSS increases by 3.61% and 30.41%, respectively, compared to the state-of-the-art method CLR and the best two-step strategy method LSR on these eight benchmark data sets. Besides the observation mentioned above, we have other observations listed in the following.

First, all the one-step clustering method, *i.e.,* LSS and CLR, outperform the two-step clustering methodes, *i.e.,* LRR, SSC, LSR and. For example, ACC values of the less effective method CLR are on average 14.30%, 22.47%, 1.09% and 6.48% higher than those of the two-step clustering methodes LRR, SSC, LSR and SMR, respectively. The reason may be that the one-step clustering methods can simultaneously obtain the ideal graph matrix and optimal clustering results, whereas the two-step clustering methodes usually generate sub-optimal solution.

Second, one-step based methods consistently outperform classic clustering methodes like K-means, RCut and NCut. For example, the less effective one-step based method CLR increased by 8.49%, 16.16%, 0.98%, and 2.43% on average, as compared to K-means, RCut and NCut on Coil20, Ecoli, Glass and Usps, respectively.

Third, the majority of the two-step clustering methods perform better than the classic clustering method K-means. For example, two-step clustering methods LRR, SSC and SMR are superior to K-means in terms of both ACC and NMI. This implies that two-step clustering methods can find correlations of data points for constructing the graph matrix, which improves the clustering performance.

4.4 Parameters' sensitivity

We vary parameter α and γ in the range [10^{-3}, ..., 10^{3}], and record ACC and NMI values of our LSS in Fig. 4. There are some important observations. First, the proposed LSS method is sensitive to the parameters, meaning that performance of LSS largely depends on parameter combinations. Actually, γ is utilized to tune the sparsity of the transfer matrix $\mathbf{W}$. Different γ value results in different level of sparsity of $\mathbf{W}$, *i.e.,* different percentage of noise features are removed from the original features. On the other hand, α is used to tradeoff the importance of $\mathbf{S}$ and $\mathbf{A}$. Second, different datasets need different range of parameters to achieve the best performance. For example, LSS achieves the best ACC (96.71%) and NMI (96.23%) on dataset Jaffe when $\gamma = 100$ and $\alpha = 10$. In contrast, LSS achieves the best ACC (83.33%) and NMI (84.52%) on Ecoli when γ and α remain the same. This indicates that our LSS is data-driven. Third, from Figure 4 we can see that parameter γ is less sensitive than α on these benchmark datasets.

4.5 Convergence

Fig. 5 shows the trend of objective value generated by our proposed Algorithm 1 with respect to iterations. Also, we set the stopping criteria of both Algorithm 1 and Algorithm 3 to $\frac{\|obj(t+1)-obj(t)\|_2^2}{obj(t)} \leq 10^{-3}$, where $obj(t)$ represents the objective function value of Eq. (5) after the t-th iteration.

From Fig. 5, we can see that our Algorithm 1 monotonically decreases the objective function value until it converges, when applying it to optimize the proposed objective function in Eq. (5). It is worth noting that the convergence rate of our Algorithm 1 is relatively fast, converging to the optimal value within 20 iterations on all the datasets used.

5 Conclusion

This paper proposes a novel Low-rank Sparse subspace clustering method via learning the affinity matrix from the low-dimensional feature space of the original data. Moreover, we integrate the learning of two affinity matrices, the learning of the transformation matrix, and the learning of the clustering results into a framework to iteratively update the affinity matrix and the transformation matrix, so that an optimal clustering results can be obtained. Extensive experimental results on both synthetic and real datasets show that the proposed LSS method outperforms the competitor methods on clustering task, and LSS has a fast convergence rate, *i.e.,* it will produce the final optimal results very quickly.

Although our proposed LSS method can obtain the significant clustering results, the number of clusters K needs to be pre-specified by the user. Similar to all K-means based methods, this is the main limitation of our LSS method. Hence, we will focus on automatically learning the value of K in our future work according to the framework of robust statistics.

References

[1] X. Zhu, S. Zhang, R. Hu, Y. Zhu *et al.*, "Local and global structure preservation for robust unsupervised spectral feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 517–529.

[2] J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *CVPR*, 2014, pp. 328–335.

[3] K. Wagstaff, C. Cardie, and S. Rogers, "Constrained k-means clustering with background knowledge," in *ICML*, 2001, pp. 577–584.

[4] W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," *IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics*, vol. 37, no. 5, p. 1382, 2007.

[5] Belongie, Serge, Carson, Hayit, Malik, and Jitendra, *Color- and Texture-Based Image Segmentation Using EM and Its Application to Content-Based Image Retrieval*, 1998.

[6] O. Uncu, W. A. Gruver, D. B. Kotak, D. Sabaz, Z. Alibhai, and C. Ng, "Gridbscan: Grid density-based spatial clustering of applications with noise," in *SMC*, 2007, pp. 2976–2981.

[7] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 3241–3248, 2011.

[8] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

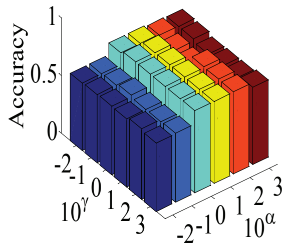
[9] C. Y. Lu, H. Min, Z. Q. Zhao, L. Zhu, D. S. Huang, and S. Yan, *Robust and Efficient Subspace Segmentation via Least Squares Regression*. Springer Berlin Heidelberg, 2012.

[10] A. Khachatryan, E. Mller, C. Stier, and K. Bhm, "Improving accuracy and robustness of self-tuning histograms by subspace clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2377–2389, 2015.

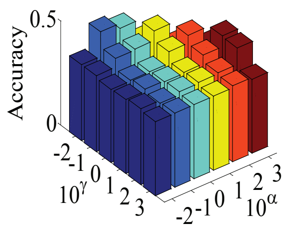
[11] X. Zhu, H.-I. Suk, L. Wang, S.-W. Lee, D. Shen, A. D. N. Initiative *et al.*, "A novel relational regularization feature selection method for joint regression and classification in ad diagnosis," *Medical image analysis*, vol. 38, pp. 205–214, 2017.

TABLE 3
Performance of all the methods on eight benchmark datasets.

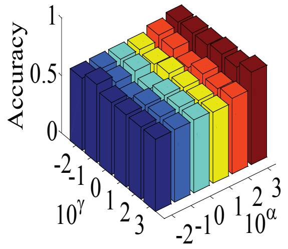
		K-means	RCut	NCut	DBSCAN	LRR	SSC	LSR	SMR	CLR	ImDB	Our LSS
ACC	Coil20	60.56	82.29	82.29	80.00	70.90	74.31	56.74	75.00	83.54	61.81	**87.15**
	Ecoli	63.99	65.48	60.42	61.90	59.23	53.57	54.46	60.71	79.46	45.54	**83.33**
	Umist	48.00	70.43	71.48	34.96	50.09	69.22	41.22	60.00	**73.22**	52.35	72.00
	Glass	52.34	50.00	51.87	35.98	50.93	52.80	50.94	50.47	52.38	42.99	**53.27**
	Usps	78.70	78.10	81.60	35.60	81.70	75.40	60.90	83.7	81.90	48.40	**84.80**
	Jaffe	81.69	85.92	88.73	77.64	85.45	94.84	54.93	92.02	85.35	82.16	**96.71**
	TOX	58.23	65.38	65.89	29.24	71.50	79.28	70.35	81.17	85.24	29.82	**87.26**
	Yeast	51.37	62.12	68.51	37.26	69.22	73.19	69.58	75.33	78.25	24.33	**80.46**
NMI	Coil20	75.08	86.81	86.81	**89.23**	78.16	87.88	64.78	80.06	85.90	75.40	88.06
	Ecoli	53.27	83.33	83.32	31.56	82.44	36.55	77.38	54.53	82.14	35.72	**84.52**
	Umist	66.08	86.65	84.12	37.43	66.08	79.09	53.40	75.41	**87.53**	74.02	81.66
	Glass	35.74	40.46	38.75	2.25	54.21	35.98	38.28	37.30	52.21	46.51	**55.61**
	Usps	67.42	82.80	83.00	31.60	70.35	72.49	62.10	74.73	82.30	54.63	**90.50**
	Jaffe	86.15	88.94	88.77	79.95	86.33	93.8	65.59	89.77	94.50	90.19	**96.23**
	TOX	56.63	61.76	61.17	3.39	68.37	73.31	69.85	76.26	79.24	16.78	**82.34**
	Yeast	59.18	67.93	69.50	3.41	73.23	75.8	73.62	80.18	83.50	15.13	**85.69**



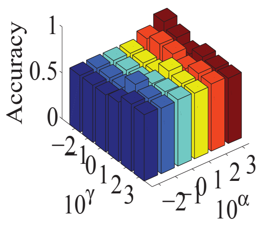
(a) Umist



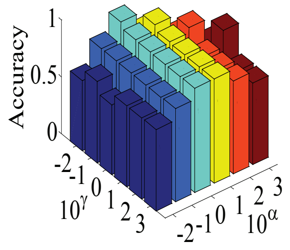
(b) Glass



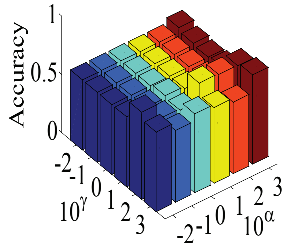
(c) Ecoli



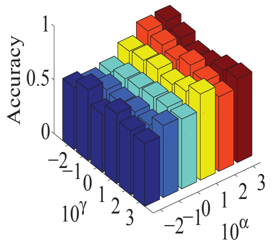
(d) Usps



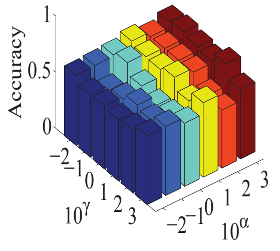
(e) Jaffe



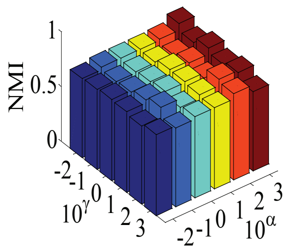
(f) Coil20



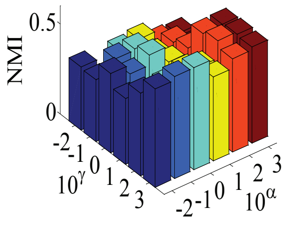
(g) Tox



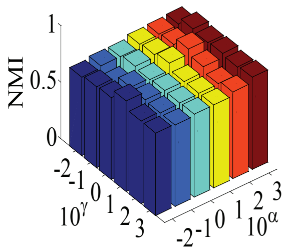
(h) Yeast



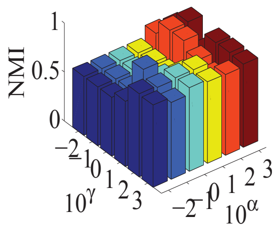
(i) Umist



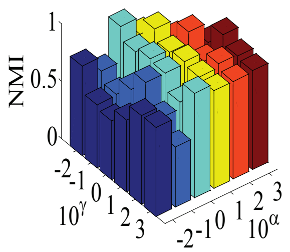
(j) Glass



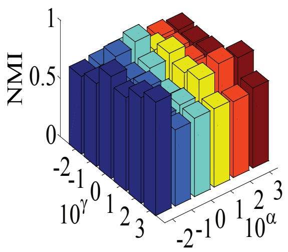
(k) Ecoli



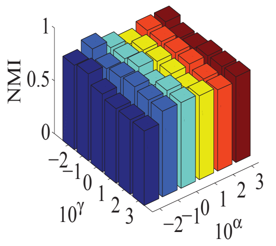
(l) Usps



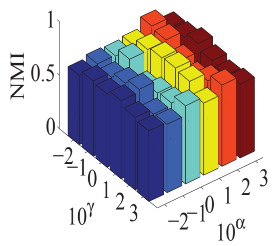
(m) Jaffe



(n) Coil20

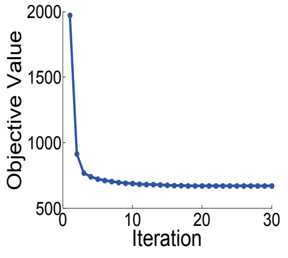


(o) Tox

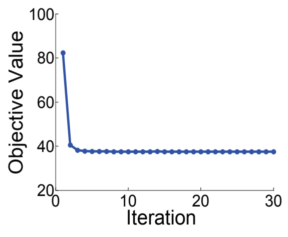


(p) Yeast

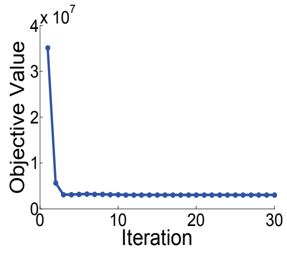
Fig. 4. ACC and NMI of our LSS method with respect to different parameter settings



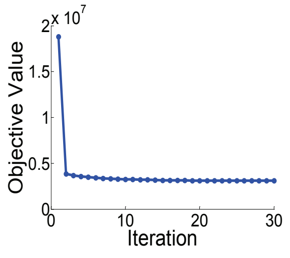
(a) Umist dataset



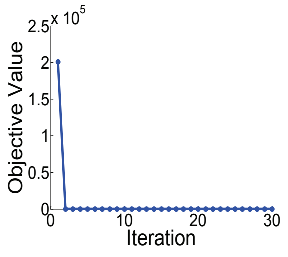
(b) Glass dataset



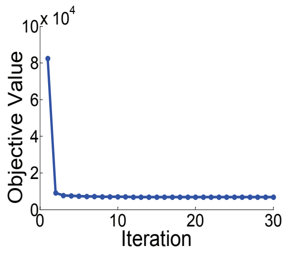
(c) Ecoli dataset



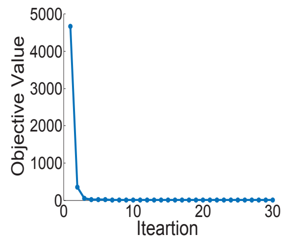
(d) Usps dataset



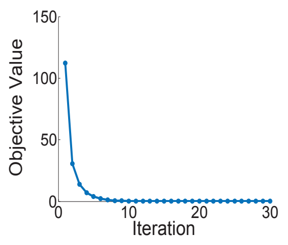
(e) Jaffe dataset



(f) Coil20 dataset



(g) Tox dataset



(h) Yeast dataset

Fig. 5. Objective value of the proposed objective function versus iteration on four benchmark datasets.

[12] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, and S. Zhang, "Graph self-representation method for unsupervised feature selection," *Neurocomputing*, vol. 220, pp. 130–137, 2017.

[13] E. Elhamifar and R. Vidal, "Sparse subspace clustering: algorithm, theory, and applications." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[14] B. Peng, L. Zhang, and D. Zhang, "A survey of graph theoretical approaches to image segmentation," *Pattern Recognition*, vol. 46, no. 3, pp. 1020–1038, 2013.

[15] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multilabel learning for image classification." *IEEE Transactions on Cybernetics*, vol. 46, no. 2, p. 450, 2016.

[16] C. Lei and X. Zhu, "Unsupervised feature selection via local structure learning and sparse learning," pp. https://doi.org/10.1007/s11 042–017–5381–7, 11 2017.

[17] 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. 171–84, 2013.

[18] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *AAAI*, 2016, pp. 1969–1976.

[19] W. Zheng, X. Zhu, Y. Zhu, R. Hu, and C. Lei, "Dynamic graph learning for spectral feature selection," *Multimedia Tools and Applications*, pp. https://doi.org/10.1007/s11 042–017–5272–y, 2017.

[20] H. Hu, Z. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *CVPR*, 2014, pp. 3834–3841.

[21] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient knn classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, 2018.

[22] X. Zhu, Z. Huang, H. T. Shen, J. Cheng, and C. Xu, "Dimensionality reduction by mixed kernel canonical correlation analysis," *Pattern Recognition*, vol. 45, no. 8, pp. 3003–3016, 2012.

[23] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for knn classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 3, p. 43, 2017.

[24] X. Zhu, L. Zhang, and Z. Huang, "A sparse embedding and least variance encoding approach to hashing," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3737–3750, 2014.

[25] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *KDD*, 2014, pp. 977–986.

[26] S. Madan and K. J. Dana, "Modified balanced iterative reducing and clustering using hierarchies (m-birch) for visual clustering," *Pattern Analysis and Applications*, vol. 19, pp. 1–18, 2015.

[27] M. K. Alazzawi, J. Luo, and R. Li, "Eecr: Energy efficient clustering using representatives for wireless sensor networks," *Journal of Computational and Theoretical Nanoscience*, vol. 12, no. 10, pp. 3516–3526, 2015.

[28] M. Sofka, J. Wetzl, N. Birkbeck, J. Zhang, T. Kohlberger, J. Kaftan, J. Declerck, and S. K. Zhou, "Multi-stage learning for robust lung segmentation in challenging ct volumes," *MICCAI*, vol. 6893, no. Pt 3, pp. 667–674, 2011.

[29] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, p. 443, 1999.

[30] A. Ram, A. Sharma, A. S. Jalal, A. Agrawal, and R. Singh, "An enhanced density based spatial clustering of applications with noise," in *IACC*, 2009, pp. 1475–1478.

[31] J. Sander, *Density-Based Clustering*. Springer US, 2011.

[32] W. Zheng, X. Zhu, G. Wen, Y. Zhu, H. Yu, and J. Gan, "Unsupervised feature selection by self-paced learning regularization," *Pattern Recognition Letters*, p. https://doi.org/10.1016/j.patrec.2018.06.029, 2018.

[33] E. Tu, L. Cao, J. Yang, and N. Kasabov, "A novel graph-based k-means for nonlinear manifold clustering and representative selection," *Neurocomputing*, vol. 143, pp. 109–122, 2014.

[34] F. Lauer and C. Schnorr, "Spectral clustering of linear subspaces for motion segmentation," in *ICCV*, 2009, pp. 678–685.

[35] X. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu, and C. Wang, "Graph pca hashing for similarity search," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2033 – 2044, 2017.

[36] S. Wu, X. Feng, and W. Zhou, "Spectral clustering of high-dimensional data exploiting sparse representation vectors," *Neurocomputing*, vol. 135, no. C, pp. 229–239, 2014.

[37] S. Wang, F. Chen, and J. Fang, "Spectral clustering of high-dimensional data via nonnegative matrix factorization," in *IJCNN*, 2015, pp. 1–8.

[38] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding." *Science*, vol. 290, no. 5500, p. 2323, 2000.

[39] M. Belkin and P. Niyogi, "Niyogi, p.: Laplacian eigenmaps for dimensionality reduction and data. neural computation 15, 1373-1396," *Neural Computation*, vol. 15, no. 6, pp. 1373—1396, 2003.

[40] X. Zhu, Z. Huang, Y. Yang, H. T. Shen, C. Xu, and J. Luo, "Self-taught dimensionality reduction on the high-dimensional small-sized data," *Pattern Recognition*, vol. 46, no. 1, pp. 215–229, 2013.

[41] X. Zhu, Z. Huang, H. Cheng, J. Cui, and H. T. Shen, "Sparse hashing for fast multimedia search," *ACM Transactions on Information Systems*, vol. 31, no. 2, p. 9, 2013.

[42] K. Fan, "On a theorem of weyl concerning eigenvalues of linear transformations i," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, no. 1, pp. 652–655, 1949.

[43] F. R. K. Chung, "Spectral graph theory cbms series," *American Mathematical Society*, vol. 9, no. 6, p. 55, 1997.

[44] B. Mohar, Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk, "The laplacian spectrum of graphs," in *Graph Theory, Combinatorics, and Applications*, 1991, pp. 871–898.

[45] I. Daubechies, R. Devore, and M. Fornasier, “Iteratively reweighted least squares minimization for sparse recovery,” *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2008.
[46] Z. Wen and W. Yin, “A feasible method for optimization with orthogonality constraints,” *Mathematical Programming*, vol. 142, no. 1-2, pp. 397–434, 2013.
[47] J. Huang, F. Nie, and H. Huang, “A new simplex sparse learning model to measure data similarity for clustering,” in *IJCAI*, 2015, pp. 3569–3575.
[48] L. Hagen and A. B. Kahng, *New spectral methods for ratio cut partitioning and clustering*. IEEE Press, 2006.
[49] C. Rate and C. Retrieval, “Columbia object image library (coil-20),” *Computer*, 2011.
[50] D. B. Graham and N. M. Allinson, *Characterising Virtual Eigensignatures for General Purpose Face Recognition*. Springer Berlin Heidelberg, 1998.
[51] J. J. Hull, “A database for handwritten text recognition research,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 2002.
[52] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Coding facial expressions with gabor wavelets,” in *AFG*, 1998, pp. 200–205.
[53] E. Y. Kwon, S. K. Shin, Y. Y. Cho, U. J. Jung, E. Kim, T. Park, J. H. Park, J. W. Yun, R. A. Mcgregor, and Y. B. Park, “Time-course microarrays reveal early activation of the immune transcriptome and adipokine dysregulation leads to fibrosis in visceral adipose depots during diet-induced obesity.” *Bmc Genomics*, vol. 13, no. 1, p. 450, 2012.
[54] S. Kaski and J. Peltonen, “Informative discriminant analysis,” in *AAAI*, 2003, pp. 329–336.
[55] J. A. Hartigan and M. A. Wong, “A k-means clustering algorithm,” *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.
[56] T. Luo, X. Zheng, G. Xu, K. Fu, and W. Ren, “An improved dbscan algorithm to detect stops in individual trajectories,” *ISPRS International Journal of Geo-Information*, vol. 6, no. 3, p. 63, 2017.

Shichao Zhang is a China 1000-Plan distinguished professor with the Guangxi Normal University, China. His research interests include data mining and partitioning. He is a senior member of the IEEE and a member of the ACM.

Xiaofeng Zhu is a faculty member at Guangxi Normal University, China. His current research interests include large-scale multimedia retrieval, feature selection, sparse learning, data preprocess, and medical image analysis.

Yonggang Li is research assitant at Guangxi Normal University, China. His current research interests include data mining and pattern recognition.

Jilian Zhang is a faculty member of College of Cyber Security, Jinan University, Guangzhou China. His current research interests include data management and data mining.

Yue Fang is a master student at Guangxi Normal University, China. His current research interests include data mining and pattern recognition.

Lifeng Yang is with Guangxi Normal University, China. His current research interests include data mining and pattern recognition.