Low-Rank Representation with Adaptive Graph Regularization

Jie Wen[a,b], Xiaozhao Fang[c], Yong Xu[a,b,*], Chunwei Tian[a,b], Lunke Fei[d]

[a]Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, Guangdong, China
[b]Shenzhen Medical Biometrics Perception and Analysis Engineering Laboratory, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, Guangdong, China
[c]School of Automation, Guangdong University of Technology, Guangzhou 510006, Guangdong, China
[d]School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, Guangdong, China

Abstract

Low-rank representation (LRR) has aroused much attention in the community of data mining. However, it has the following two problems which greatly limit its applications: 1) it cannot discover the intrinsic structure of data owing to the neglect of the local structure of data; 2) the obtained graph is not the optimal graph for clustering. To solve the above problems and improve the clustering performance, we propose a novel graph learning method named low-rank representation with adaptive graph regularization (LRR_AGR) in this paper. Firstly, a distance regularization term and a non-negative constraint are jointly integrated into the framework of LRR, which enables the method to simultaneously exploit the global and local information of data for graph learning. Secondly, a novel rank constraint is further introduced to the model, which encourages the learned graph to have very clear clustering structures, *i.e.*, exactly c connected components for the data with c clusters. These two approaches are meaningful and beneficial to learn the optimal graph that discovers the intrinsic structure of data. Finally, an efficient iterative algorithm is provided to optimize the model. Experimental results on synthetic and real datasets show that the proposed method can significantly improve the clustering performance.

Keywords: Low-rank representation, graph regularization, data clustering, rank constraint

1. Introduction

In the fields of machine learning and pattern recognition, data analysis technologies can be generally divided into three groups in view of whether use the label information during model training, *i.e.*, supervised learning, unsupervised learning, and semi-supervised learning [13, 51, 43, 50, 45, 30, 55]. With the development of computer devices and internet, unsupervised learning arouses more and more attention since data are usually very large and their labels are difficult to be obtained. For example, there are more than 5 billion photos in Flickr and 13.7 PB photos in Google photo albums. How to automatically manage these large amounts of photos into different

[*] Corresponding author
Email address: `yongxu@ymail.com` (Yong Xu)

natural groups is a very challenging problem. As an unsupervised learning method, data clustering is the most favored technique to deal with this challenge [14, 59]. Data clustering aims at discovering the natural groups of data without any label information. In the past decades, various clustering methods have been proposed and can be categorized into two groups: hierarchical clustering and partitional clustering [13]. Single-link, average-link, and complete-link based methods are the most typical hierarchical clustering methods which produce a set of nested clusters via a hierarchical tree [33]. However, it is unrealistic to calculate the hierarchical tree for large-scale data. Compared with hierarchical clustering, partitional clustering can simultaneously partition data into different groups which are much preferred in researches and applications [53]. In this branch, the most popular methods are K-means clustering [16], density based clustering [38], and spectral clustering [35, 52], etc. K-means clustering seeks for a partition that minimizes the within-cluster distances. Based on the assumption that data points located in a high density region have larger possibility to be a group, density based clustering methods determine the cluster structure by searching for the connected dense regions of data. Compared with K-means and density based clustering methods which are usually directly performed on the sample space, spectral clustering is more flexible since it is performed on a meaningful low dimensional representation derived from the original data.

Spectral clustering can be regarded as the graph-based clustering since its performance is directly determined by the obtained graph. A good graph should reveal the intrinsic relationships among samples. Two popular assumptions are widely used in graph construction:

1) **Distance assumption** [1]: Samples with small distance should have larger possibility to be a cluster.

2) **Representation assumption** [4]: Sample y can be efficiently represented by a linear combination of other samples in dataset X, and those samples from the same subspace with sample y have more contribution in the representation.

The first assumption usually uses Euclidean distance as the measure metric to construct the graph. Following the first assumption, ratio cut (Rcut) [8] and normalized cut (Ncut) [42] construct a similarity graph, where all nodes are calculated via Gaussian kernel function. Belkin et al. learned a k nearest neighbor graph (*knn*-graph) with k non-zero nodes for each point [42]. Based on the representation assumption, Roweis et al. constructed a local linear embedding graph (*LLE*-graph) which first finds k nearest neighbors for each sample and then calculates the linear representation coefficients of these k nearest neighbors [41]. *knn*-graph reveals the distance relationships among nearest neighbor samples while *LLE*-graph captures the representation relationships between sample and its nearest neighbor samples. Although the above two graphs are beneficial to learn a compact low-dimensional representation for clustering, their performances are sensitive to the neighbor number k. In addition, the above methods are also sensitive to the presence of noises and outliers. Thus it is urgent to propose a method that can learn a graph with adaptive neighbors for any data. To this end, sparse subspace clustering (SSC) is proposed [4]. Compared with *LLE*-graph, SSC can adaptively and flexibly select few samples via l_1 norm sparisty constraint rather than Euclidean distance to select k samples for graph construction. Besides, SSC can reduce the negative influence of outliers and noises by introducing a sparse error term into graph learning model. These factors enable SSC to perform better than the *LLE*-graph and *knn*-graph based methods. However, for the case that two samples have same similarity degree as the represented sample, SSC fails because it will only select one of them for representation while ignoring another one due to the property of l_1 norm [62].

The above graph learning methods including SSC only aim at capturing the local structure of data while ignoring the global structure. To capture the global structure of data, many methods have been proposed, among which low-rank representation (LRR) is the most popular [22, 27, 19, 60]. By imposing a low-rank constraint on the representation matrix, LRR jointly learns a representation graph that can exactly uncover the intrinsic subspace structures. Based on LRR, Liu et al. further proposed the latent low-rank representation (LatLRR) method to deal with the insufficient sampling problem and improve the robustness to noise by using the 'hidden information' of data [23]. However, LRR and LatLRR only capture the global representation structure while ignoring the local structure of data. Generally, local structure also contains lots of discriminative information [40]. And thus exploiting both two information may be beneficial to find the true clusters of data. Inspired by this motivation, various extensions of LRR have been proposed [61, 56]. For example, Zhuang et al. learned a non-negative low-rank and sparse (NNLRS) graph by jointly introducing the low-rank and sparse constraints to regularize the representation matrix [61]. Sparse constraint allows the graph learned by NNLRS to capture the local linear representation structure of data. Besides, many researchers propose to impose the Laplacian regularizer on the representation matrix to exploit the local information of data, which enforces similar samples to have similar representations [56, 24, 3].

For data with c clusters, the ideal graph for clustering should better have exactly c connected components [6, 37]. However, this structure cannot be always guaranteed by the previous graph learning methods in most cases, especially for data with dependent subspaces [6]. To solve this problem, some researchers propose to use the clusters information as prior for graph learning [6, 37]. For example, Feng et al. imposed the rank constraint on the Laplacian matrix of the representation graph to enforce the learned graph to have exactly c connected components [6]. Reference [37] proves that the optimal graph can be obtained from a fixed graph by simply imposing the rank constraint. Using cluster information as prior for graph learning is reasonable since data are expected to be exactly divided into c clusters in many cases. Inspired by this motivation, in this paper, we propose a novel robust graph learning method named low-rank representation with adaptive graph regularization (LRR_AGR) for data clustering. Different from the method proposed in [37] which learns the ideal graph from a pre-defined graph, LRR_AGR seeks to adaptively learn such ideal graph from data, thus it is possible to obtain the global optimal graph for clustering. Different from the methods proposed in [6] and [37] which only capture the global or local structure of data, LRR_AGR integrates the distance regularization into the framework of low-rank representation to simultaneously capture the global and local structures of data, which encourages the obtained graph to discover the intrinsic structure of data. By introducing the regularization of rank constraint, LRR_AGR can adaptively learns a graph with exactly c connected components, which makes the obtained graph be more suitable for data clustering task. In summary, the proposed method has the following advantages in comparison with the other methods:

(1) LRR_AGR simultaneously exploits the global representation information and local distance information, which enables LRR_AGR to learn the optimal graph that captures the intrinsic relationships of data.

(2) By introducing a novel rank constraint, LRR_AGR has potential to learn a graph with clear

clustering structure, which encourages the method to obtain a better performance.

(3) The proposed method has the potential to adaptively select the exact nearest neighbors for graph construction without manual intervention, which greatly improves the adaptability in the real-world clustering applications including the case of data with manifold structure.

(4) The obtained graph has good interpretability, in which each element directly reveals the similarity relationship of the corresponding two samples.

We conduct several experiments on the synthetic and public benchmark datasets to evaluate the effectiveness of the proposed method. Experimental results show that the proposed method can significantly improve the clustering performance.

The paper is organized as follows: In section 2, we give a brief review to several related works. Section 3 presents the proposed method and its optimal solution. Section 4 analyzes the proposed method from aspects of computational complexity and convergence property, etc. Section 5 conducts several experiments. Section 6 offers the conclusion of the paper.

2. Related works

In this section we briefly introduce two graph-based clustering methods, *i.e.*, sparse subspace clustering (SSC) [4] and low-rank representation (LRR) [22] which are the most related methods to the proposed method.

2.1. Sparse subspace clustering (SSC)

SSC uses the sparse representation technique to adaptively select few samples rather than *k* nearest neighbors for graph construction. For a data matrix $X = [x_1, x_2, \ldots, x_n] \in R^{m \times n}$ with *n* samples, SSC attempts to solve the following objective function to obtain the sparse representation graph for data clustering

$$\min_{Z} \|Z\|_1 \; s.t. \; X = XZ, diag\,(Z) = 0 \tag{1}$$

where $Z = [z_1, z_2, \ldots, z_n] \in R^{n \times n}$ is the sparse representation graph and each column z_i is the representation coefficient vector corresponding to sample x_i. $\|\cdot\|_1$ is the l_1 norm constraint and is calculated as $\|Z\|_1 = \sum\limits_{i,j}^{n} |z_{ij}|$ [31]. Due to the sparsity selection property of l_1 norm, some elements of each column z_i will be forced to zero [54]. $diag(Z) = 0$ means that the diagonal elements of matrix Z are enforced to zero. Most importantly, due to the competitiveness property in joint representation, samples from the same subspace as the sample to be represented are more possible to be selected for representation. Thus those samples with nonzero representation values are more possible to be the same cluster as the sample to be represented. This ensures the graph obtained by SSC to adaptively capture the intrinsic local representation geometric structure of data.

Caused by the inaccurate data collection techniques, data may be corrupted by noises and sparse outliers in the real-world applications [15]. To reduce the negative influence of the above corruptions, Elhamifar et al. further extended SSC to the following graph learning model [4]

$$\begin{gathered}\min_{Z,E,A} \|Z\|_1 + \lambda_1 \|E\|_1 + \frac{\lambda_2}{2} \|A\|_F^2 \\ s.t.\ X = XZ + E + A, diag\left(Z\right) = 0, \mathbf{1}^T Z = \mathbf{1}^T\end{gathered} \tag{2}$$

where *E* denotes the sparse outliers, *A* is noises. $\mathbf{1} = [1, 1, \ldots, 1]^T$ is a column vector with all elements are 1. $\|\cdot\|_F$ is the Frobenius norm and defined as $\|X\|_F = \sqrt{\sum_{i,j} x_{i,j}^2}$ [25]. X^T denotes the transposed matrix of X.

After obtaining the representation graph *Z*, SSC first normalizes each column of *Z* as $z_i = z_i / \|z_i\|_\infty$, then calculates the similarity graph $W = |Z| + |Z|^T$ whose each element w_{ij} denotes the similarity degree between samples x_i and x_j. Based on similarity graph *W*, SSC finally utilizes spectral clustering [6] to achieve the clustering result.

2.2. Low-rank representation (LRR)

Compared with SSC, LRR seeks to jointly learn a representation graph with the lowest rank for data clustering [22]. The general model of LRR is formulated as

$$\min_{Z} rank\left(Z\right)\ \ s.t.\ X = XZ \tag{3}$$

where $rank\left(Z\right)$ is the rank of matrix *Z*. Problem (3) is NP-hard, and difficult to optimize. To address this issue, many researchers transform the rank minimization problem into the following nuclear norm based minimization problem [22]

$$\min_{Z} \|Z\|_* \ s.t.\ X = XZ \tag{4}$$

where $\|Z\|_*$ is the nuclear norm constraint of matrix *Z* and is calculated as $\|Z\|_* = \sum_i^n \delta_i$, δ_i is the *i*-th singular value of matrix *Z*. Similar to SSC, LRR is also extended to the following learning model for image with noises and gross corruptions

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_p \ s.t.\ X = XZ + E \tag{5}$$

where *E* is the error term used to model different noises or outliers with different norm constraints *p*, such as l_1 norm and $l_{2,1}$ norm, etc.

Problem (5) can be efficiently solved by using the exact or inexact ALM (IALM) algorithm [21]. After obtaining *Z*, LRR also calculates similarity graph *W* like SSC and then applies the spectral clustering algorithm to segment data into different subspaces.

3. Low-Rank Representation with Adaptive Graph Regularization

As analyzed in the previous section, SSC and LRR only aim at learning the representation graph that uncovers the representation relationships of samples while ignoring the local distance relationships. Besides, the representation coefficients of each sample cannot clearly show the similarity degree between samples since many representation values are negative. Therefore, graphs obtained by these two methods do not have good interpretability and cannot reveal the intrinsic structure of data. In this paper, we focus on learning a more general graph which holds the following properties for clustering:

(1) The obtained graph should capture both local and global structures of data.

(2) All elements of the obtained graph should be non-negative so that they can directly reveal the similarity degree of samples.

(3) The obtained graph should have exactly connected structures.

3.1. Model of LRR_AGR

Recent years, many researches have shown the importance of locality preserving [57, 48, 44, 28, 10, 29]. However, this property is ignored in many LRR based graph learning methods. Although NNLRS [61] uses the sparse constraint of l_1 norm to adaptively select few samples for data representation, it also cannot capture the intrinsic locality structure of data because sparsity does not necessarily guarantee the locality [48]. To guarantee the locality, in this paper, we introduce a simple distance constraint and a non-negative constraint rather than the sparsity constraint into the graph leaning model of LRR as follows

$$\min_{Z} \sum_{i,j}^{n} \|x_i - x_j\|_2^2 z_{ij} + \lambda_1 \|Z\|_* \ s.t.\ X = XZ, Z \geq 0 \tag{6}$$

where $X = [x_1, x_2, \ldots, x_n] \in R^{m \times n}$ is the data matrix with *n* samples, in which each sample is represented by a column vector. $Z \in R^{n \times n}$ is the representation graph needs to be learned, each element z_{ij} denotes the representation coefficient of sample x_j in the joint representation with respect to sample x_i. λ_1 is a positive penalty parameter. $Z \geq 0$ means that all elements of Z are non-negative. By jointly introducing the two constraints, model (6) holds the following good properties:

- Introducing the non-negative constraint has several good properties: (1) it avoids the undesired solution, *i.e.*, any two non-nearest neighbor samples are connected by a larger negative coefficient; (2) the obtained graph directly reveals the similarity degree between samples; (3) learning a non-negative graph has the potential to obtain a better performance [9].

- The first regularization term, *i.e.*, $\sum\limits_{i,j}^{n} \|x_i - x_j\|_2^2 z_{ij}$ can be viewed as the weighted sparse regularization when Z is non-negative, which enables the method to adaptively select few nearest neighbor samples for representation. And thus introducing this constraint can simultaneously guarantee the locality and sparsity.

- Model (6) simultaneously exploits the global and local information of data for graph construction, which encourages the method to learn the optimal graph that captures the intrinsic structure of data.

To eliminate the influence of self-representation, we further introduce a constraint that enforce the diagonal elements of graph to zero as follows:

$$\min_{Z} \sum_{i,j}^{n} \|x_i - x_j\|_2^2 z_{ij} + \lambda_1 \|Z\|_* \; s.t. \; X = XZ, diag\left(Z\right) = 0, Z \geq 0 \tag{7}$$

In the real-world applications, data may be corrupted by noises. To reduce the negative influence of noises, a sparse error term is introduced as follows

$$\begin{gathered} \min_{Z,E} \sum_{i,j}^{n} \|x_i - x_j\|_2^2 z_{ij} + \lambda_1 \|Z\|_* + \lambda_2 \|E\|_1 \\ s.t. \; X = XZ + E, diag\left(Z\right) = 0, Z \geq 0 \end{gathered} \tag{8}$$

where *E* denotes the error, λ_2 is also a positive penalty parameter.

Although model (8) can simultaneously capture the local and global structures of data, and is robust to noise to some extent, it is not the best graph for data clustering. This is mainly because model (8) cannot ensure the obtained graph to have exactly *c* connected components for data with *c* clusters. To learn such ideal graph, references [6, 37] integrate a novel rank constraint into the graph learning model based on the following theorem.

Theorem 1: Let *Z* be a non-negative affinity matrix and its Laplcian matrix L_Z is defined as $L_Z = D - \left(Z + Z^T\right)/2$, where D is a diagonal matrix and its *i*-th diagonal element $D_{ii} = \sum_j \left(z_{ij} + z_{ji}\right)/2$. Then multiplicity *k* of eigenvalue 0 of Laplacian matrix L_Z equals to the number of connected components of affinity matrix *Z* [47].

Based on **Theorem 1**, the problem to learn the ideal graph for data with *c* clusters is transformed into the following optimization problem

$$\begin{gathered} \min_{Z,E} \sum_{i,j}^{n} \|x_i - x_j\|_2^2 z_{ij} + \lambda_1 \|Z\|_* + \lambda_2 \|E\|_1 \\ s.t. \; X = XZ + E, diag\left(Z\right) = 0, Z \geq 0, rank\left(L_Z\right) = n - c \end{gathered} \tag{9}$$

To avoid the extreme case that elements of any row of graph *Z* are all zero, we further introduce a constraint to enforce the sum of each row of Z to 1. As a result, the final non-negative graph learning model of the proposed method is as follows

$$\begin{gathered} \min_{Z,E} \sum_{i,j}^{n} \|x_i - x_j\|_2^2 z_{ij} + \lambda_1 \|Z\|_* + \lambda_2 \|E\|_1 \\ s.t. \; X = XZ + E, diag\left(Z\right) = 0, Z \geq 0, rank\left(L_Z\right) = n - c, \sum\nolimits_j z_{ij} = 1 \end{gathered} \tag{10}$$

By introducing the novel rank constraint, model (10) can adaptively learn such optimal graph with exactly *c* connected components. After obtaining the affine graph, we finally perform normalized cut (Ncut) algorithm[1] [42] to obtain the clustering results.

3.2. Solution to the LRR_AGR

In this section, we mainly present the solution to the proposed graph learning model. As the Laplacian matrix L_Z is positive semi-definite, then all eigenvalues of L_Z should be equal to or greater than 0. Therefore, the optimization problem (10) is equivalent to the following minimization problem [37, 36]

$$\begin{gathered}\min_{Z,E} \sum_{i,j}^{n} \|x_i - x_j\|_2^2 z_{ij} + \lambda_1 \|Z\|_* + \lambda_2 \|E\|_1 + 2\lambda_3 \sum_{i}^{c} \sigma_i\left(L_Z\right) \\ s.t. X = XZ + E, diag\left(Z\right) = 0, Z \geq 0, \sum\nolimits_j z_{ij} = 1\end{gathered} \tag{11}$$

where $\sigma_i\left(L_Z\right)$ denotes the *i*-th smallest eigenvalue of L_Z and $\sigma_i\left(L_Z\right) \geq 0$. λ_3 is a positive penalty parameter. When λ_3 is large enough, the forth term of problem (11) should be enforced to zero. In this case, the first *c* smallest eigenvalues of L_Z are enforced to zero and thus the rank constraint $rank\left(L_Z\right) = n - c$ is satisfied [37, 36].

Problem (11) is still difficult to be directly solved. Thanks to the Theorem proposed by Fan [5], we can further rewrite problem (11) into the following equivalent optimization problem:

$$\begin{gathered}\min_{Z,E,F} \sum_{i,j}^{n} \|x_i - x_j\|_2^2 z_{ij} + \lambda_1 \|Z\|_* + \lambda_2 \|E\|_1 + 2\lambda_3 Tr\left(F^T L_Z F\right) \\ s.t.\ X = XZ + E, diag\left(Z\right) = 0, Z \geq 0, \sum\nolimits_j z_{ij} = 1, F^T F = I\end{gathered} \tag{12}$$

where $F = \left[f_1^T, f_2^T, \ldots, f_n^T\right]^T \in R^{n \times c}$, *n* and *c* are the number of samples and clusters of data, respectively. $Tr(\cdot)$ is the trace operator.

Compared with the optimization problem (10), (12) is much easier to be solved by many methods, such as accelerated proximal gradient (APG) [46] and alternating direction method (ADM) [11], etc. In view of the efficiency of ADM, we choose it to optimize problem (12). We first introduce two variables, *i.e.*, S and U, to make (12) separable for optimization as follows

$$\begin{gathered}\min_{Z,S,U,E,F} \sum_{i,j}^{n} \|x_i - x_j\|_2^2 s_{ij} + \lambda_1 \|U\|_* + \lambda_2 \|E\|_1 + 2\lambda_3 Tr\left(F^T L_S F\right) \\ s.t.\ X = XZ + E, Z = S, Z = U, diag\left(S\right) = 0, S \geq 0, \sum\nolimits_j s_{ij} = 1, F^T F = I\end{gathered} \tag{13}$$

[1] Code of Ncut is available at: http://www.cis.upenn.edu/ jshi/software/

Then we rewrite (13) into the following augmented Lagrangian formula [51]

$$\begin{aligned} L(Z,S,U,E,F) = &\sum_{i,j}^{n} \|x_i - x_j\|_2^2 s_{ij} + \lambda_1 \|U\|_* + \lambda_2 \|E\|_1 + 2\lambda_3 Tr\left(F^T L_S F\right) + \langle C_1, X - XZ - E\rangle \\ &+ \langle C_2, Z - S\rangle + \langle C_3, Z - U\rangle + \frac{\mu}{2}\left(\|X - XZ - E\|_F^2 + \|Z - S\|_F^2 + \|Z - U\|_F^2\right) \end{aligned} \tag{14}$$

where C_1, C_2, and C_3 are Lagrange multipliers, μ is a positive penalty parameter. By alternately solving each variable of (14) with other variables fixed, we can obtain the solution of all variables Z, S, U, E, F. The detail solution steps are as follows.

Step 1. Update Z: When variables S, U, E, F are fixed, Z can be obtained by minimizing the following formula:

$$L(Z) = \left\|X - XZ - E + \frac{C_1}{\mu}\right\|_F^2 + \left\|Z - S + \frac{C_2}{\mu}\right\|_F^2 + \left\|Z - U + \frac{C_3}{\mu}\right\|_F^2 \tag{15}$$

By setting the derivative $\partial L(Z)/\partial Z = 0$, we can obtain variable Z as follows

$$Z = \left(X^T X + 2I\right)^{-1} \left(X^T L_1 + L_2 + L_3\right) \tag{16}$$

where $L_1 = X - E + \frac{C_1}{\mu}$, $L_2 = S - \frac{C_2}{\mu}$, and $L_3 = U - \frac{C_3}{\mu}$.

Step 2. Update S: Fixing variables Z, U, E, F, we can obtain S by minimizing the following problem:

$$\begin{aligned} &\min_S \sum_{i,j}^{n} \|x_i - x_j\|_2^2 s_{ij} + 2\lambda_3 Tr\left(F^T L_S F\right) \ + \frac{\mu}{2}\left\|Z - S + \frac{C_2}{\mu}\right\|_F^2 \\ &\qquad\qquad s.t.\ diag\left(S\right) = 0, S \geq 0, \sum\nolimits_j s_{ij} = 1 \end{aligned} \tag{17}$$

From (17), we have

$$\begin{aligned} &\sum_{i,j}^{n} \|x_i - x_j\|_2^2 s_{ij} + \lambda_3 \sum_{i,j}^{n} \|f_i - f_j\|_2^2 s_{ij} + \frac{\mu}{2}\left\|Z - S + \frac{C_2}{\mu}\right\|_F^2 \\ =&\sum_{i,j}^{n} \left(\|x_i - x_j\|_2^2 + \lambda_3 \|f_i - f_j\|_2^2\right) s_{ij} + \frac{\mu}{2}\left\|Z - S + \frac{C_2}{\mu}\right\|_F^2 \\ =&\sum_{i,j}^{n} g_{ij} s_{ij} + \frac{\mu}{2}\|S - H\|_F^2 \\ =&Tr\left(G^T S\right) + \frac{\mu}{2}\|S - H\|_F^2 \end{aligned} \tag{18}$$

where each element of G is calculated by $g_{ij} = \|x_i - x_j\|_2^2 + \lambda_3 \|f_i - f_j\|_2^2$, $H = Z + \frac{C_2}{\mu}$. From (17) and (18), it is obvious to see that problem (17) is equivalent to solving the following problem

$$\min_{s_i \geq 0, s_i 1 = 1, s_{ii} = 0} \|s_i - (h_i - g_i/\mu)\|_2^2 \tag{19}$$

where s_i, h_i, g_i denote the *i*-th row of S, H, and G, respectively. Problem (19) has a closed form solution and can be fast solved by an efficient algorithm presented in reference [37].

Step 3. Update F: When Z, S, U, E are fixed, F can be obtained by solving the following minimization problem:

$$F = \arg\min_{F} Tr\left(F^T L_S F\right) \; s.t. \; F^T F = I, F \in R^{n\times c} \tag{20}$$

where L_S is the Laplacian matrix of S. Problem (20) can be simply solved via eigenvalue decomposition and its solution is the set of *c* eigenvectors corresponding to the first *c* smallest eigenvalues of L_S.

Step 4. Update U: U can be achieved by solving the following problem with variables Z, S, E, F fixed

$$U = \arg\min_{U} \lambda_1 \|U\|_* + \frac{\mu}{2}\left\|Z - U + \frac{C_3}{\mu}\right\|_F^2 \tag{21}$$

Then U is obtained as follows by using the singular value thresholding (SVT) operator [23]

$$U = \Theta_{\lambda_1/\mu}\left(Z + \frac{C_3}{\mu}\right) \tag{22}$$

where Θ denotes the SVT operator.

Step 5. Update E: E is obtained by solving the following minimization problem

$$E = \arg\min_{E} \lambda_2 \|E\|_1 + \frac{\mu}{2}\left\|X - XZ - E + \frac{C_1}{\mu}\right\|_F^2 \tag{23}$$

E has the following closed solution

$$E = \Omega_{\lambda_2/\mu}\left(X - XZ + C_1/\mu\right) \tag{24}$$

where Ω is the shrinkage operator [21].

Step 6. Update C_1, C_2, C_3, μ: Lagrange multipliers C_1, C_2, C_3, and penalty parameter μ are respectively updated by using the following formulas:

$$C_1 = C_1 + \mu(X - XZ - E) \tag{25}$$

$$C_2 = C_2 + \mu\left(Z - S\right) \tag{26}$$

$$C_3 = C_3 + \mu\left(Z - U\right) \tag{27}$$

$$\mu = \min(\rho\mu, \mu_{\max}) \tag{28}$$

where ρ and $\mu_{\max}$ are constants.

The proposed optimization approach is summarized in Algorithm 1. After obtaining similarity graph Z, we further use Ncut to achieve the final clustering result.

Algorithm 1 LRR_AGR (solving (11))

```
Input: Data $X$; parameters $\lambda_1, \lambda_2, \lambda_3$; cluster number $c$.
Initialization: Constructing the $k$ nearest neighbor graph as the initial matrix of $Z$; $U = Z$, $S = Z$; using (21) to calculate the initial matrix of $F$; $C_1 = C_2 = C_3 = 0$, $E = 0$, $\mu = 0.01$, $\rho = 1.2$, $\mu_{\max} = 10^8$.
while not converged do
    1. Update $Z$ by using (16).
    2. Update $S$ by solving (19).
    3. Update $F$ by solving (20).
    4. Update $U$ by using (22).
    5. Update $E$ by using (24).
    6. Update $C_1, C_2, C_3, \mu$ by using (25), (26), (27), and (28), respectively.
end while
Output: $Z, S, U, E, F$
```

4. Analysis of the proposed method

4.1. Computation complexity and convergence analysis

For LRR_AGR listed in Algorithm 1, the most computational costs are the inverse operation, eigen-decomposition, and singular value thresholding (SVT) in steps 1, 3, and 4, respectively. For a matrix with the size of $n \times n$, the computational complexities of inverse operation, eigen-decomposition, and SVT are $O\left(n^3\right)$, $O\left(n^2 c\right)$ and $O\left(n^3\right)$, respectively, where c is the number of the selected eigenvectors. Thus the computational complexities of step 1, step 3, and step 4 are about $O(n^3)$, $O\left(n^2 c\right)$, and $O\left(n^3\right)$, respectively. Note that we do not take into account the basic matrix operations, such as matrix addition, subtraction, and multiplication. Considering that the inverse operation of $\left(X^T X + 2I\right)^{-1}$ in step 1 can be pre-computed and utilized in all iteration steps, thus the total computational complexity of the proposed method is about $O\left(n^3 + \tau\left(n^2 c + n^3\right)\right)$, where τ is the iteration number.

From the above analyses, we can find that step 4 has the highest computational complexity in algorithm 1, which needs $O\left(n^3\right)$ to implement the SVT operation. This may limit the applications to the data with large amounts of samples. In fact, the major computational cost in SVT operation is the singular value decomposition (SVD). One of the widely used approaches to improve the efficiency of step 4 is to exploit the partial SVD via PROPACK [17], which has the computational complexity of $O\left(rn^2\right)$, where r is the rank of matrix $(Z + C_3/\mu)$. However, when rank $r > n/5$, the computational cost of using PROPACK is usually much higher than computing the full SVD [2]. To address this issue, we can exploit a more efficient approach proposed in [2] to improve the efficiency of step 4, which mainly contains the following three steps:

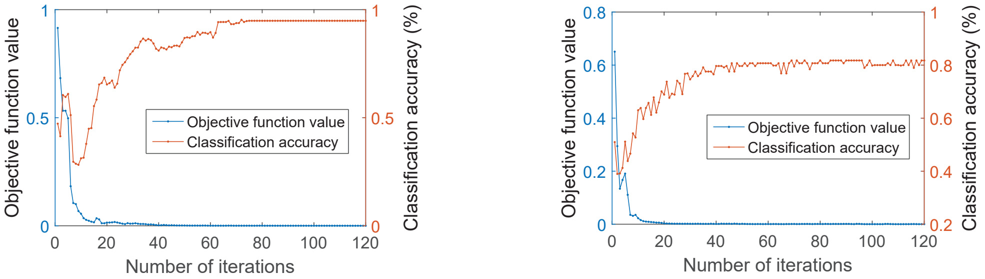
(1) Denote $Y = Z + C_3/\mu$, then obtain a unitary matrix W and a symmetric nonnegative definite matrix Q via the polar decomposition $Y = WQ$;

(2) Project matrix Q into a 2-norm ball by optimizing problem $\mathrm{P}_{\lambda_1/\mu}\left(Q\right) := \mathop{\arg\min}\limits_{\|X\|_2 \le \lambda_1/\mu} \|X - Q\|_F$;

(3) Obtain matrix U as $U = Y - W\mathrm{P}_{\lambda_1/\mu}(Q)$.

Compared with the full SVD and partial SVD, this approach can be fast computed by the basic linear algebra subroutine (BLAS) without calculating the singular value decomposition, and thus can greatly improve the efficiency of the proposed method in large scale datasets.

As presented in the previous section, we utilize the ADM-style method to iteratively achieve the solution. In [22, 21], the convergence property of ADM with two blocks has been proved. However, it is unrealistic to prove the strong convergence for the method with five blocks. In this section, we use some experiments to prove the convergence property of the proposed method. Fig.1 shows the objective function value and clustering accuracy with respect to the number of iterations. It should be noted that the objective function value of the proposed method is calculated as $Obj = (\sum\limits_{i,j}^{n} \|x_i - x_j\|_2^2 z_{ij} + \lambda_1 \|Z\|_* + \lambda_2 \|E\|_1 + \|X - XZ - E\|_F^2)/\|X\|_F$. From Fig.1, it is obvious to see that the objective function value decreases sharply in the first few steps and then tends to be smooth. Meanwhile, the clustering accuracy increases till the peak point and then also tends to smooth. It should be pointed out that owing to the sensitivity of K-means to the initialization, the clustering accuracy presented in Fig.1(b) has small fluctuations. This case is acceptable. Above analyses and Fig.1 indicate that after a few iteration steps, the proposed method can converge to the local optimal solution.



(a) COIL20

(b) Umist

Figure 1: Objective function values and clustering accuracy versus the number of iteration of the proposed method on the COIL20 and Umist datasets, in which all classes of each dataset are selected in the experiments.

4.2. *Connections to other methods*

In this section, we mainly analyze the connections among the proposed method and some related methods, such as LRR [22], SSC [4], NNLRS [61], CAN [37], low-rank matrix factorization with adaptive graph regularizer (LRMF_AGR) [26], graph regularized compact LRR (GCLRR) [3], and Laplacian regularized LRR (LapLRR) [24].

(1) Connections to LRR, SSC, and NNLRS: Based on the representation assumption presented in the introduction, SSC and LRR respectively use the sparsity technique and low-rank constraint to learn a graph that captures the structure information of data. However, these two methods only preserve one type of structures of data, in which SSC only captures the local representation structure while LRR only preserves the global representation structure of data. This indicates that graphs learned by SSC and LRR cannot reveal the intrinsic structure of data. In addition, elements

of graphs obtained by these two methods only denote the representation contribution of two samples. In other words, it cannot demonstrate the probability of two samples to be in the same cluster. Thus graphs obtained by SSC and LRR do not have good interpretability. Compared with LRR and SSC, NNLRS solves the above problems by learning a non-negative graph with properties of sparsity and low-rank. NNLRS can be viewed as a more general method which integrates LRR and SSC into a joint learning framework. However, NNLRS only focuses on the representation structure while ignoring the nearest neighbor structure of data. To solve the above problems, we integrate a distance regularization term instead of the sparsity constraint into the graph learning model of LRR. The distance regularization term encourages the nearest neighbor samples to have larger representation weight in the joint representation. Compared with NNLRS which only preserves the sparsity property to select few samples for representation, the proposed method can simultaneously preserve sparsity and locality by using the distance regularization term [48]. In addition, another difference compared with NNLRS is that LRR_AGR imposes a novel rank constraint on the graph. By introducing the rank constraint, LRR_AGR can adaptively learn an ideal graph with clear connected structures, which is beneficial to obtain a better clustering performance. Both distance regularization constraint and rank constraint are useful and encourage LRR_AGR to obtain the optimal graph. Several experiments in the following section also demonstrate the effectiveness of the proposed method.

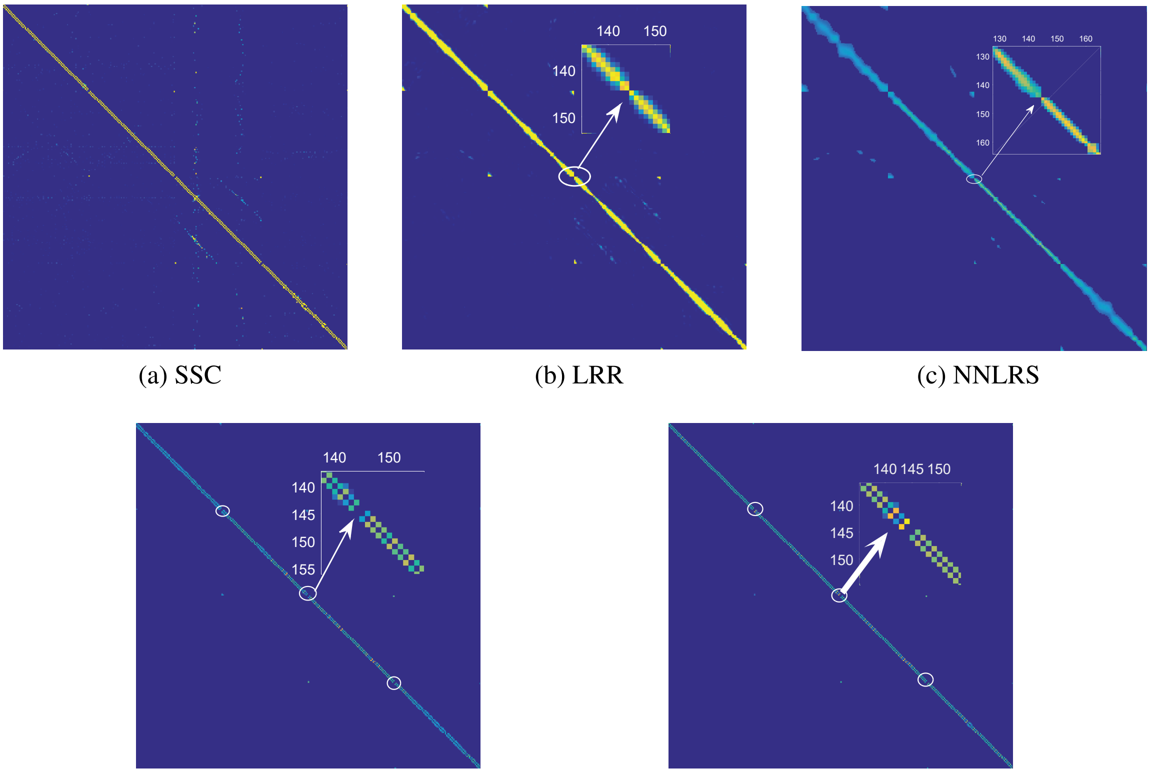
(2) Connections to CAN: CAN exploits the distance relationships of samples to adaptively learn a graph Z with exactly c connected components as follows [36]:

$$\begin{gathered} \min_{Z} \sum_{i,j}^{n} \left(\|x_i - x_j\|_2^2 z_{ij} + \gamma z_{ij}^2 \right) \\ s.t. \forall i, z_i^T \mathbf{1} = 1, 0 \leq z_i \leq 1, rank\left(L_Z\right) = n - c \end{gathered} \tag{29}$$

where $z_i \in R^{n\times 1}$ is a column vector, $\mathbf{1} \in R^{n\times 1}$ is a column vector with all elements are 1, γ is the regularization parameter. $L_Z = D - \frac{Z^T + Z}{2}$ is the Laplacian matrix, where D is a diagonal matrix and its i-th element is calculated as $D_{ii} = \sum_j \left(z_{ij} + z_{ji}\right)/2$.

From (10) and (29), if we replace the nuclear norm constraint with the l_2 norm and delete the self-representation constraint, then LRR_AGR degrades to CAN. So CAN can be viewed as a special case of LRR_AGR to some extent. In the real-world applications, samples always distribute in the nonlinear manifold [41]. In this case, CAN fails because Euclidean distance cannot reveal the intrinsic similarity relationships of samples in the nonlinear manifold. To tackle this bottleneck, we introduce a low-rank representation constraint to capture this nonlinear intrinsic structure of data. Therefore, the proposed method is a more general model than CAN in dealing with the linear and nonlinear cases. In addition, Euclidean distance is sensitive to the corruption of noise, which indicates that CAN fails to capture the intrinsic structure hidden behind noisy data. While the proposed method has the potential to learn a latent clear graph by introducing the low-rank representation and utilizing a sparse error term to compensate noises, which guarantees the method to achieve a better performance than CAN.

Fig.2 shows the graphs obtained by SSC, LRR, NNLRS, CAN, and the proposed method on the subset of COIL20 dataset, respectively. Clustering accuracies corresponding to these graphs are shown in Table 4. It is obvious to see that there are many ‘messy points’ in the first three graphs respectively obtained by SSC, LRR, and NNLRS, which confuse the intrinsic subspace structure of graph and thus lead to an unsatisfactory clustering performance. Compared with LRR, SSC, and NNLRS, graphs obtained by CAN and the proposed method do not have any redundant ‘scatter point’ and have very clear connected structures which are partitioned by three gaps marked by ‘white ellipse’. This indicates that using the cluster information as prior knowledge is beneficial to adaptively learn a graph with ideal cluster structure so as to obtain a better clustering performance.



(a) SSC

(b) LRR

(c) NNLRS

(d) CAN

(e) LRR_AGR

Figure 2: Graphs obtained by SSC, LRR, NNLRS, CAN, and the proposed method, respectively. Note: (1) In this experiments, images of the first four classes of the COIL20 dataset are selected. (2) For the last four graphs, the local area marked by the ‘ellipse’ is amplified and the corresponding amplified sub-image is pointed by the direction of ‘arrow’.

(3) Connections to LRMF_AGR, GCLRR, and LapLRR. The learning models of the three methods are respectively expressed as follows:

$$LRMF\_AGR: \min_{U,V,Z} \left\|X - UV^T\right\|_F^2 + \lambda_1 Tr\left(V^T L_Z V\right) + \|X - XZ\|_F^2 + \lambda_2 \|Z\|_F^2 + \lambda_3 \|Z\|_1 \quad (30)$$

$$GCLRR: \min_{W,H,E} \|Z\|_* + \lambda_1 \|E\|_{2,1} + \lambda_2 Tr\left(ZLZ^T\right) \; s.t. \, X = XWZ + E, W^T W = I \quad (31)$$

$$LapLRR : \min_{Z\geq 0} \frac{1}{2} \left\| X - XZ \right\|_F^2 + \lambda_1 \|Z\|_* + \frac{\lambda_2}{2} Tr \left(ZLZ^T \right) \tag{32}$$

where L_Z denotes the Laplacian matrix of graph Z, L is the Laplacian matrix of the pre-constructed nearest neighbor graph. We can find that the above three methods and the proposed method all exploit the graph regularizer to improve their performances. To summarize the proposed learning model in (10) and the above three models, we can obtain that: 1) LRMF_AGR is obviously different because it seeks to learn a more compact clustering indictor matrix with low-dimension, while other three methods focus on finding the intrinsic graph of data for clustering. 2) LRMF_AGR only captures the local information of data while ignoring the global structure of data. Compared with LRMF_AGR, the other methods all simultaneously take into account the global and local structures of data, which is beneficial to obtain a better performance. 3) GCLRR and LapLRR use the same Laplacian term to exploit the local information of data. Both of two methods expect the similar samples to have similar representation vectors. One of shortcomings of the two methods is that they are sensitive to the selection of the nearest neighbor numbers [26]. Different from G-CLRR and LapLRR, the proposed method uses a unique approach to exploit the local information of data, which expects the nearest samples to have more contributions in the joint representation. From the learning model (10), we can find that the proposed method does not need to set the number of nearest neighbors. Moreover, by introducing the rank constraint, the proposed method has the potential to learn the graph with clear cluster structure while the other methods cannot, which further enables the proposed method to perform better than GCLRR and LapLRR.

5. Experiments and Analysis

In this section, we conduct several experiments on both synthetic dataset and real benchmark datasets, to evaluate the proposed method. Then we make a discussion for the LRR_AGR method. All experiments are performed on the software Matlab 2015b and Windows 10 system, hardware Intel Core i7-4790 CPU and 16GB ram. The code of the proposed method is available at: http://www.yongxu.org/lunwen.html.

Following related clustering methods are compared with the proposed method:

(1) *K*-means clustering method
(2) Ratio cut (Rcut) clustering method [41]
(3) Normalized cut (Ncut) clustering method[2] [42]
(4) Sparse subspace clustering (SSC) method[3] [41]
(5) LRR clustering method[4] [22]
(6) LatLRR clustering method [23]
(7) Non-negative low-rank and sparse (NNLRS) graph based clustering method [61]
(8) Laplacian regularized LRR (LapLRR) [24]
(9) Non-negative sparse hyper-Laplacian regularized LRR (NSHLRR)[5] [56]

[2] Code of Ncut is available at: http://www.cis.upenn.edu/ jshi/software/

[3] Code of SSC is available at: http://www.vision.jhu.edu/code/

[4] Code of LRR is available at: http://www.cis.pku.edu.cn/faculty/vision/zlin/zlin.htm

[5] Code of NSHLRR is available at: `https://www.researchgate.net/profile/Ming_Yin3`

(10) Clustering with adaptive neighbors (CAN)[6] [36]

(11) Constraint Laplacian rank (CLR) clustering method [37]

Among the above compared methods, *K*-means is directly performed on the original features while the others are performed on different graphs learned from data. In the experiments, Rcut and Ncut are performed on the adjacency graph constructed by Gaussian kernel [11]. SSC, LRR, LatLRR, NNLRS, LapLRR, and NSHLRR perform the spectral clustering (SC) [35] on the obtained graphs for data clustering. Compared with all above methods, CAN and CLR directly use the obtained similarity graph to partition data rather than use the SC method. In addition, CAN and CLR use the cluster number as prior for clustering. Considering that the above methods are sensitive to the parameters to some extent, we perform these algorithms with a wide parameter range and report their best results for fair comparison. For example, for methods like Rcut and Ncut, we select the nearest neighbor number of *k* from a candidate domain of $\{3, 5, 7, 11, 13, 15, 20, 25\}$ and select the scale value of Gaussian kernel from a candidate domain of $\{1, 3, ..., 23\}$ to construct different adjacency graphs for data clustering and then report the best clustering result. For CAN, we utilize the method proposed by the authors to construct the initial graph for data clustering [37]. It should be noted that the above methods except CAN and CLR all use *K*-means to segment data into respective groups. So we perform *K*-means 10 times and report the mean values for fair comparison in terms of the sensitivity of *K*-means to the initialization.

5.1. Evaluation metrics

In this paper, we adopt two metrics, *i.e.*, clustering accuracy (Acc) and normalized mutual information (NMI) [12] to evaluate the clustering performance of different algorithms. For a dataset $X = [x_1, x_2, \ldots, x_n] \in R^{m\times n}$ with *n* samples, Acc is calculated as follows

$$Acc = \frac{\sum_{i=1}^{n} \delta\left(map\left(r_i\right), y_i\right)}{n} \tag{33}$$

where r_i and y_i denote the cluster label obtained by the clustering algorithm and the true label of sample x_i, respectively. Permutation mapping function $map\left(\cdot\right)$ is used to map each prediction cluster label r_i to the equivalent label according to the distribution of the true label [56]. Function $\delta\left(x, y\right)$ equals one under the condition of $x = y$ and equals zero otherwise.

After obtaining the predicted cluster label *R*, NMI is defined as follows [56]

$$NMI\left(R, Y\right) = \frac{MI\left(R, Y\right)}{\max\left(H\left(R\right), H\left(Y\right)\right)} \tag{34}$$

where *Y* is the true label of data, $H\left(R\right)$ and $H\left(Y\right)$ are the entropy of labels *R* and *Y*, respectively. Mutual information $MI\left(\cdot\right)$ is calculated as follows

$$MI\left(R, Y\right) = \sum_{s\in R}\sum_{t\in Y} p\left(s, t\right)\log_2\left(\frac{p\left(s, t\right)}{p\left(s\right)p\left(t\right)}\right) \tag{35}$$

[6] Codes of CAN and CLR are available at: http://www.escience.cn/people/fpnie/index.html

Table 1: Clustering mean Accs (%) and NMIs (%) and their standard deviations of different methods on the two-moon synthetic dataset. Note: bold numbers denote the best results.

Method	ACC	NMI
K-means	51.50±0.00	0.08±0.00
Rcut	58.89±3.45	2.99±1.76
Ncut	58.50±0.00	2.89±1.75
SSC	67.00±0.00	8.51±0.00
LRR	67.00±0.00	8.51±0.00
LatLRR	67.00±0.00	8.51±0.00
NNLRS	67.00±0.00	15.65±0.00
LapLRR	59.50±0.26	2.47±0.14
NSHLRR	67.00±0.00	8.51±0.00
CAN	64.00	48.00
CLR	60.00	20.50
LRR_AGR	**90.85±0.34**	**60.68±0.70**

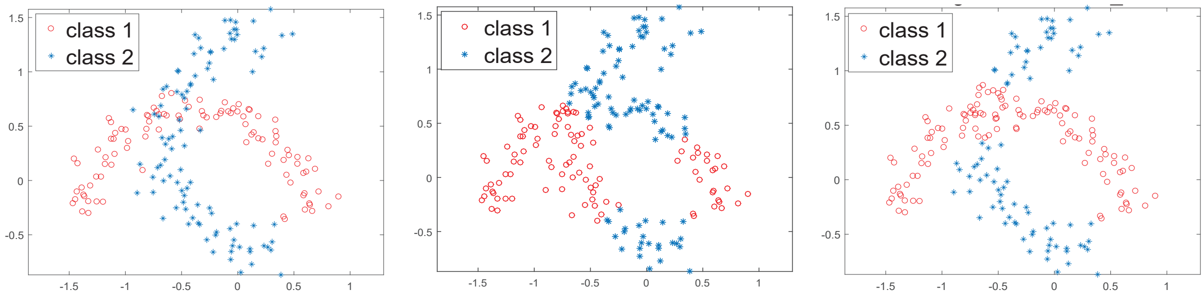
where $p(s,t)$ denotes the joint probability distribution of s and t, $p(s)$ and $p(t)$ are the marginal probability of s and t, respectively. The maximum and minimum value of Acc and NMI are 1 and 0, respectively. Generally, the larger the value of Acc or NMI is, the better the clustering performance is.

5.2. Experiments on synthetic dataset

In this section, we choose the two-moon synthetic dataset to measure the clustering performance of the proposed method. Fig.3(a) shows the two-moon dataset which has two natural clusters of data distributed in the moon shape. Fig.3(b) and Fig.3(c) show the clustering results obtained by LRR and the proposed method, respectively. We can find that the proposed method significantly outperforms LRR on this dataset. In addition, we also compare the clustering Acc and NMI with the other methods and show the clustering results in Table 1. It is obvious to see that the proposed method obtains the best performance in comparison with the other methods, where the clustering Acc of the proposed method is 23% higher than the second best methods, *i.e.*, SSC, LRR, LatLRR, NNLRS, and NSHLRR. Moreover, Table 1 also indicates that only preserving the nearest neighbor structure or representation structure cannot well describe the intrinsic distribution of data and thus cannot obtain satisfactory clustering performance.

5.3. Experiments on real datasets

In this section, we conduct experiments on some real datasets, including four non-image datasets from UCI machine learning repository [20] and six image datasets, to further prove the effectiveness of the proposed method. Table 2 shows the description of the used datasets. The Extended Yale B (YaleB) [20] face dataset contains 2414 images provided by 38 persons under different illumination conditions. The original AR face dataset [32] contains over 4000 frontal face images corresponding to 126 people under different facial expressions, illumination conditions, and occlusions by sun glasses and scarf. We choose a subset of AR dataset with 3120



(a) Original two-moon dataset (b) Results of LRR (c) Results of LRR_AGR

Figure 3: Results of LRR and LRR_AGR on the two-moon synthetic dataset.

Table 2: Description of datasets.

Dataset	No. of instances	Dimensions	Classes
YaleB [7]	2414	1024	38
AR [32]	3120	2000	120
Umist [39]	575	2576	20
LFW [18]	1251	1024	86
MSRA	1799	256	12
COIL20 [34]	1440	1024	20
Cars [20]	392	8	3
Vehicle [20]	846	18	4
Isolet [20]	1560	617	2
Yeast [20]	1484	8	10

images from 120 individuals to evaluate different clustering methods. The Umist face dataset[7] [39] contains 575 images from 20 volunteers. Images of each class have different poses from profile to frontal views. The Labeled Faces in the Wild (LFW) face dataset [18] contains more than 13000 images collected from the web. In the experiments, we use a subset of LFW which contains 1251 face images from 86 people for the comparison of different methods [49]. The Columbia Object Image Library (COIL20) dataset[8] [34] is an object dataset which is composed of 1440 gray-scale images of 20 classes. We use the processed images without background and resize all images to the size of 32 × 32. MSRA, Cars, Vehicle, Isolet, and Yeast datasets are available at http://www.escience.cn/people/fpnie/papers.html, in which MSRA is a face dataset and the others are non-image datasets from UCI machine learning repository [20]. For YaleB, COIL20, and Umist datasets, we conduct a series of experiments with a range of first c sub-clusters which are selected from these datasets for the comparison of different methods. For the remaining seven datasets, all samples are chosen to perform experiments.

Table 3-Table 6 show the clustering mean Accs (%) of different methods on the above real-world datasets. The clustering mean NMIs of these methods are shown in Fig.4. From the experimental results, we can obtain the following conclusions:

(1) For the five image datasets and Isolet dataset which have large dimension of features, the proposed method significantly outperforms the other methods in comparison of Acc and NMI. We can also find that on the YaleB and Umist datasets, the proposed method performs much

[7] The Umist dataset is available at: http://cs.nyu.edu/ roweis/data.html

[8] The COIL20 dataset is available at: http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

better than the other methods in all cases. For example, on the YaleB dataset in Table 3, the proposed method achieves more than 4.5% average scores of Acc in comparison with the second best method NNLRS. While on the remaining three non-image datasets which have limited features per sample, the proposed method achieves the comparative clustering Acc with the other methods. This indicates that more features are beneficial for the proposed method to capture the intrinsic structure of data so as to improve the clustering performance.

(2) From Table 3 and Table 5, we can find that the proposed method performs much better than CLR and CAN which also use the cluster number as prior for graph learning. Especially on the YaleB dataset with 38 clusters, the proposed method achieves more than 45% average Acc scores in comparison with CLR and CAN. This observation indicates that the global representation structure contains useful information for data clustering.

(3) In Table 4 and Table 5, compared with Rcut, Ncut, and CAN which all use the nearest neighbor information for graph learning, CAN and the proposed method perform better than Rcut and Ncut in most cases. This indicates that using the cluster number information as prior knowledge is beneficial to learn a more robust graph that better captures the intrinsic structure of data so as to improve the clustering performance.

(4) From Table 4, we can find that the clustering Accs of Rcut, Ncut, CAN, and CLR, are much higher than those of SSC, LRR and LatLRR. Compared with SSC, LRR, and LatLRR, which only capture the representation structure of data, other methods utilize the local nearest neighbor information of data for clustering. This indicates that the local distance relationships of data are also very useful and contain sufficient discriminability for data clustering in some cases.

(5) From Fig.4, we can find that, all the other methods perform much better than the conventional K-means in most cases. This phenomenon demonstrates that it is not a good choice to exploit the original data directly for data clustering because it contains many redundant features even noises. Most importantly, it is obvious to see that the proposed method achieves the highest NMIs in all datasets, which proves the effectiveness of the proposed method for data clustering tasks. Since all methods except the K-means exploit the similar clustering approach based on their obtained graphs, thus the good performance verifies that the proposed method can learn a more discriminative and reasonable graph from data.

(6) From Fig.4 and Tables 4 and 5, we can find that LapLRR and NSHLRR obtain better performance than LRR. LapLRR and NSHLRR mainly introduce the similar Laplacian term to LRR for graph learning. Thus the experimental results prove that introducing the Laplacian term has the potential to improve the clustering performance. However, from the results in Table 3, the performance of LapLRR is worse than the conventional LRR on the YaleB dataset, which indicates that the Laplacian term plays an opposite role in guiding the graph learning. The above experimental results demonstrate that the Laplacian regularized approach of LRR is really sensitive to the pre-constructed nearest neighbor graph. While from the experimental results of Accs and NMIs shown in the above tables and Fig.4, we can find that the proposed method obtains consistently better performance than LapLRR and NSHLRR. This proves that the distance constraint of the proposed method is more effective than the Laplacian term in discovering the intrinsic relationships of data.

From the above analyses, we commonly obtain that both local and global information of data are useful in data clustering. Employing both local structure and global structure of data encour-

Table 3: Clustering mean Accs (%) of different methods on the YaleB dataset. Note: bold numbers denote the best results.

Classes	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	NNLRS	LapLRR	NSHLRR	CAN	CLR	LRR_AGR
2	50.85	94.53	94.53	**100**	78.13	96.88	99.21	99.22	99.22	98.44	98.44	**100**
8	18.91	50.03	50.30	88.31	83.79	83.20	89.77	83.67	84.05	54.10	51.37	**99.61**
14	15.95	54.33	54.60	78.71	89.46	82.06	93.41	77.64	83.24	49.77	47.73	**96.76**
20	12.11	55.09	55.89	76.84	90.44	80.10	92.03	76.55	86.78	46.67	45.25	**94.19**
26	11.35	56.14	56.70	76.89	87.39	74.79	90.73	75.10	80.47	48.54	39.79	**94.05**
32	10.76	51.44	51.46	76.12	80.65	77.18	85.49	81.23	83.96	46.70	39.85	**92.82**
38	9.39	48.77	49.42	73.89	70.34	78.88	82.41	77.29	80.54	42.88	36.99	**87.04**
Avg.	18.47	58.62	58.99	81.54	82.89	81.87	90.44	81.87	85.47	55.30	51.35	**94.92**

Table 4: Clustering mean Accs (%) of different methods on the COIL20 dataset. Note: bold numbers denote the best results.

Classes	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	NNLRS	LapLRR	NSHLRR	CAN	CLR	LRR_AGR
4	62.15	86.46	82.64	62.50	96.53	91.32	98.61	96.53	98.37	**100**	**100**	**100**
6	48.36	91.68	92.05	62.70	64.12	67.59	83.80	81.90	85.48	**100**	**100**	**100**
8	43.66	86.88	86.96	77.26	70.83	65.28	80.56	74.31	78.24	**100**	**100**	**100**
10	46.06	83.82	86.04	67.11	68.47	68.22	84.17	75.01	80.17	**100**	**100**	**100**
12	53.41	83.16	81.70	79.98	62.99	66.81	84.03	78.54	83.65	**100**	**100**	**100**
14	56.81	83.01	81.97	74.01	66.36	75.00	86.31	78.98	80.24	**100**	**100**	**100**
16	60.83	82.35	79.81	75.28	69.33	71.25	83.41	77.95	81.87	**100**	**100**	**100**
18	64.96	81.09	82.62	75.53	66.54	67.18	85.26	80.94	83.74	94.29	**100**	**100**
20	57.67	76.49	77.83	77.92	66.39	65.64	80.31	75.01	81.48	90.14	87.36	**97.31**
Avg.	54.88	83.88	83.51	72.48	70.17	69.76	85.16	79.91	83.69	98.27	98.60	**99.70**

Table 5: Clustering mean Accs (%) of different methods on the Umist dataset. Note: bold numbers denote the best results.

Classes	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	NNLRS	LapLRR	NSHLRR	CAN	CLR	LRR_AGR
4	47.97	66.34	67.05	60.16	61.79	51.55	73.17	84.55	88.62	90.24	78.86	**91.71**
6	52.91	73.37	73.57	70.35	68.02	50.12	73.26	82.56	88.90	**93.02**	84.88	**93.02**
8	48.45	74.70	76.90	67.42	70.47	60.37	76.26	86.85	80.69	**94.37**	84.51	**94.37**
10	44.57	68.63	68.58	71.70	74.60	70.04	75.66	77.64	78.53	82.26	77.74	**83.47**
12	44.66	69.27	69.37	67.00	64.87	68.77	82.88	69.78	72.22	80.18	77.48	**87.39**
14	41.52	69.85	69.46	71.67	60.18	69.56	81.23	73.56	79.35	84.32	76.86	**86.12**
16	39.84	60.57	61.37	65.97	54.63	64.75	73.78	65.74	71.35	77.42	72.81	**77.51**
18	38.78	61.34	62.00	67.30	55.90	61.34	78.59	65.40	68.73	70.59	69.87	**80.45**
20	41.58	62.33	62.57	63.48	56.28	61.04	74.63	65.78	66.15	73.74	70.26	**81.65**
Avg.	44.48	67.38	67.87	67.23	62.97	61.95	76.61	74.65	77.17	82.90	77.03	**86.19**

ages the model to learn the optimal graph that uncovers the intrinsic geometric structure of data. In particular, using the cluster number as a prior knowledge makes the obtained graph have ideal connected structure which is more suitable for the clustering task. With the integration of above factors, the proposed method achieves better performance than the other methods.

In addition, we conduct a significance test to better show the statistical significance of the proposed method in comparison with the other methods [58, 19]. Table 7 listed p-values of mean clustering Accs between LRR_AGR and the other methods on the YaleB dataset, where the significance level is set as 0.05. When the estimated p-value is lower than 0.05, the performance difference between the two compared methods is statistically significant. From Table 7, it is obvious to see that the performance differences of the mean clustering Accs between the proposed method and all the other methods are statistically significant in all cases. These experimental results also verify the effectiveness of the proposed method in data clustering.

Table 6: Clustering mean Accs (%) of different methods on remaining real datasets. Note: bold numbers denote the best results.

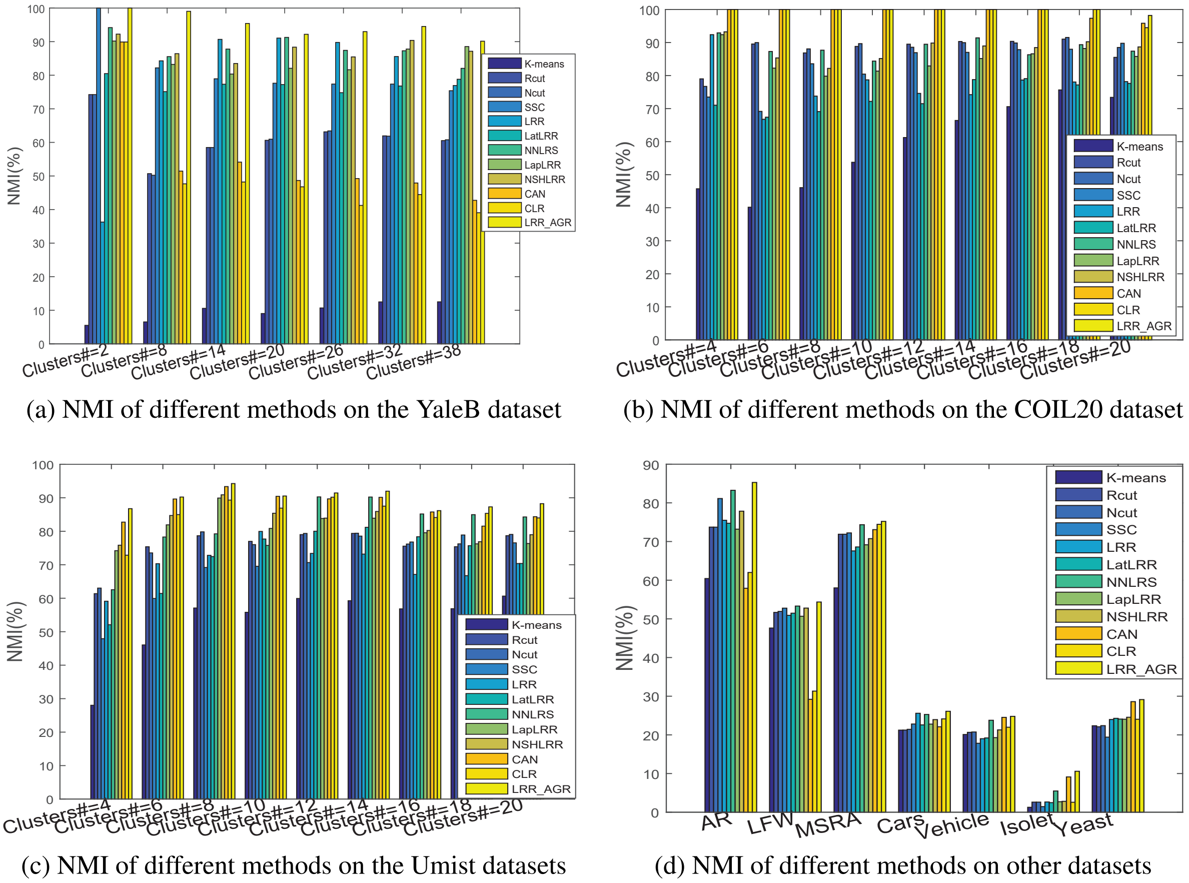
Classes	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	NNLRS	LapLRR	NSHLRR	CAN	CLR	LRR_AGR
AR	31.19	48.39	48.52	64.73	56.37	57.14	66.25	65.19	65.75	40.10	42.47	**66.74**
LFW	22.18	23.79	24.03	29.29	23.81	25.08	27.52	27.49	27.98	17.75	18.79	**29.47**
MSRA	50.70	57.42	57.42	60.98	63.79	60.57	65.78	62.89	63.41	59.14	59.14	**66.08**
Cars	54.59	63.01	63.11	62.00	62.50	67.96	68.36	64.09	62.34	**68.42**	67.86	68.11
Vehicle	45.86	46.53	46.64	44.92	45.75	46.57	46.95	45.89	45.98	46.81	45.39	**47.24**
Isolet	56.60	59.30	59.30	55.13	59.55	52.44	61.62	59.49	58.11	59.23	59.23	**62.67**
Yeast	31.79	34.42	34.67	39.34	36.93	37.40	42.19	37.42	40.74	**50.47**	45.89	50.13

Table 7: p-values of mean clustering Accs between LRR_AGR and other methods on the YaleB dataset. The asterisk '*' represents that the difference between the two methods is statistically significant when $p = 0.05$.

Methods	2	8	14	20	26	32	38
K-means	$1.0 \times 10^{-18*}$	$8.1 \times 10^{-20*}$	$3.4 \times 10^{-19*}$	$5.7 \times 10^{-20*}$	$1.3 \times 10^{-18*}$	$8.3 \times 10^{-21*}$	$5.1 \times 10^{-20*}$
Rcut	0*	$7.6 \times 10^{-14*}$	$2.9 \times 10^{-12*}$	$1.3 \times 10^{-11*}$	$5.6 \times 10^{-12*}$	$7.6 \times 10^{-13*}$	$8.4 \times 10^{-13*}$
Ncut	0*	$8.1 \times 10^{-15*}$	$1.2 \times 10^{-12*}$	$4.5 \times 10^{-13*}$	$7.8 \times 10^{-12*}$	$1.1 \times 10^{-15*}$	$8.2 \times 10^{-13*}$
SSC	–	$2.7 \times 10^{-14*}$	$2.5 \times 10^{-11*}$	$2.1 \times 10^{-7*}$	$5.3 \times 10^{-11*}$	$6.6 \times 10^{-12*}$	$2.5 \times 10^{-10*}$
LRR	0*	0*	0*	$3.9 \times 10^{-13*}$	$8.1 \times 10^{-11*}$	$1.4 \times 10^{-10*}$	$3.3 \times 10^{-11*}$
LatLRR	0*	0*	$6.3 \times 10^{-11*}$	$4.9 \times 10^{-11*}$	$4.2 \times 10^{-10*}$	$2.1 \times 10^{-8*}$	$6.4 \times 10^{-7*}$
NNLRS	0*	$1.1 \times 10^{-14*}$	$2.3 \times 10^{-4*}$	$3.3 \times 10^{-5*}$	0.008*	$1.7 \times 10^{-7*}$	$1.7 \times 10^{-7*}$
LapLRR	0*	$3.7 \times 10^{-15*}$	$7.6 \times 10^{-14*}$	$4.7 \times 10^{-13*}$	$2.5 \times 10^{-14*}$	$2.9 \times 10^{-11*}$	$1.1 \times 10^{-9*}$
NSHLRR	0*	$2.8 \times 10^{-16*}$	$1.7 \times 10^{-14*}$	$5.8 \times 10^{-13*}$	$1.7 \times 10^{-14*}$	$2.5 \times 10^{-10*}$	$1.9 \times 10^{-10*}$
CAN	0*	0*	$2.1 \times 10^{-26*}$	0*	$1.9 \times 10^{-24*}$	$3.3 \times 10^{-24*}$	$2.2 \times 10^{-17*}$
CLR	0*	0*	$1.4 \times 10^{-26*}$	0*	$4.0 \times 10^{-25*}$	$9.6 \times 10^{-25*}$	$7.1 \times 10^{-18*}$

5.4. Parameter sensitivity and selection

In this section, we use some experiments to show the sensitivity of parameters to the clustering Acc of the proposed method. From the objective function of (12), we can find that there are three regularization parameters, *i.e.*, λ_1, λ_2, and λ_3 needed to be set in advance. Parameters λ_1, λ_2, and λ_3 respectively balance the importance of low-rank constraint term, error term, and rank constraint. Generally, the larger the parameter value is, the more importance or impact of the corresponding term is. To demonstrate the effects of these three parameters for data clustering, we first define a candidate parameter range set of $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}\}$ for these three parameters and then perform the proposed method with different combinations of parameters for data clustering. We first fixed parameters λ_1 and λ_2, and then perform the proposed method with different values of parameter λ_3 to show the influence of λ_3 to the clustering Acc. From Fig.5 (a) and Fig.6 (a), it is obvious to see that the clustering Acc is insensitive to parameter λ_3 when $\lambda_3 \leq 0.01$. This is mainly because if parameter λ_3 is too large, the corresponding rank constraint term will play the dominant role in the graph learning while ignoring the local and global structure preservation. In this case, although the obtained graph still has *c* connected components, it cannot reveal the intrinsic structure of data. In the experiments, we can select a small value in the range of $\{10^{-5}, 10^{-4}, 10^{-3}\}$ for parameter λ_1. Fig.5 (b) and Fig.6 (b) show the clustering Acc versus different values of parameters λ_1 and λ_2 when parameter λ_3 is fixed. As can be seen from Fig.5 (b) and Fig.6 (b), the clustering Acc is sensitive to parameter λ_2 to some extent and the best clustering result can be obtained when parameters λ_1 and λ_2 are in a feasible range. This is mainly because a very large or very small parameter λ_2 leads to a small error or large error that cannot well compensate the sparse noise of data. In this case, the model cannot learn the intrinsic



(a) NMI of different methods on the YaleB dataset

(b) NMI of different methods on the COIL20 dataset

(c) NMI of different methods on the Umist datasets

(d) NMI of different methods on other datasets

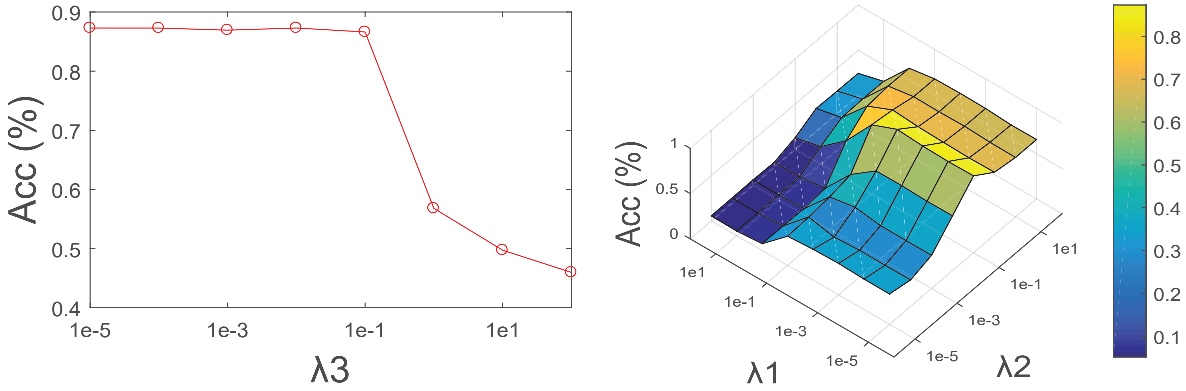
Figure 4: Mean NMIs (%) of different methods on benchmark datasets.

similarity graph for data clustering. Thus in the experiments, we can select the two parameters in the candidate range of $\{10^{-3}, 10^{-2}, 10^{-1}\}$ according to the degree of noise corruptions of data.

As far as we know, it is still an open problem to adaptively select these optimal parameters for different datasets. In the experiments, we first fix parameter λ_3 since this parameter is insensitive to the clustering Acc, and then perform the method to find the optimal λ_1 and λ_2 in a candidate domain where the optimal parameters may exist. Then by similar strategy, we fix parameters λ_1 and λ_2 to find the optimal value of parameter λ_3 in a candidate domain. Finally, the optimal combination of these parameters can be obtained in the 3D candidate space which is composed of three candidate domains of parameters.

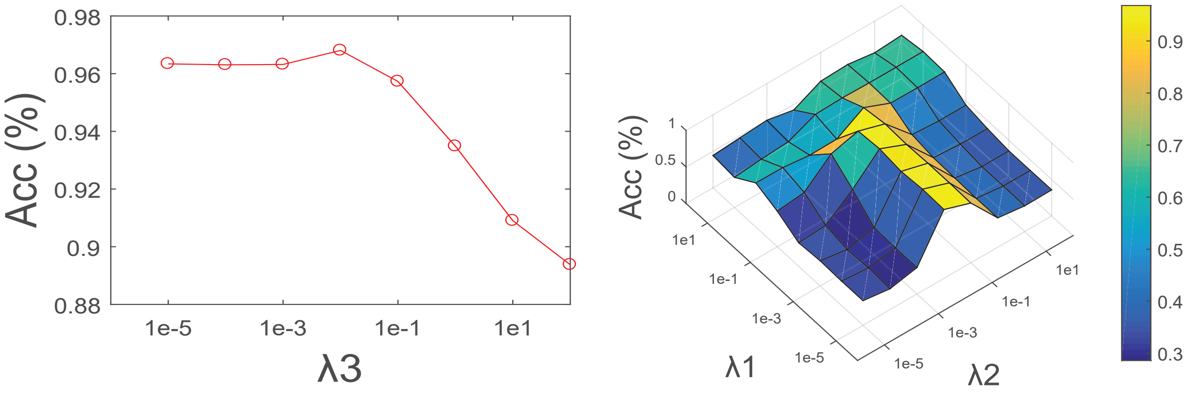
6. Conclusion

In this paper, we propose a novel graph learning method to learn a non-negative graph with clear connected structures for data clustering. In particular, a distance regularization term is integrated into the conventional low-rank representation model to exploit the local information of data for graph construction. To make the graph have an ideal cluster structure, a rank constraint is introduced to the graph learning model. Meanwhile, by introducing a non-negative constraint, the



(a) (b)

Figure 5: Clustering Acc (%) versus different values of (a) parameter λ_3, (b) parameters λ_1 and λ_2 on the YaleB dataset.



(a) (b)

Figure 6: Clustering Acc (%) versus different values of (a) parameter λ_3, (b) parameters λ_1 and λ_2 on the COIL20 dataset.

interpretability of the graph is greatly improved. Compared with the other methods, graph obtained by the proposed method not only captures the local and global intrinsic structure information of data, but also has exactly connected components for data clustering. Extensive experiments on both synthetic and real datasets show the effectiveness of the proposed method for data clustering.

Acknowledgments

This work is supported in part by Economic, Trade and Information Commission of Shenzhen Municipality (Grant no. 20170504160426188) and in part by Guangdong Province high-level personnel of special support program (Grant No. 2016TX03X164).

References

[1] Belkin, M., Niyogi, P., 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems. pp. 585–591.

[2] Cai, J.-F., Osher, S., 2013. Fast singular value thresholding without singular value decomposition. Methods and Applications of Analysis 20 (4), 335–352.

[3] Du, S., Ma, Y., Ma, Y., 2017. Graph regularized compact low rank representation for subspace clustering. Knowledge-Based Systems 118, 56–69.

[4] Elhamifar, E., Vidal, R., 2013. Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11), 2765–2781.

[5] Fan, K., 1949. On a theorem of weyl concerning eigenvalues of linear transformations i. Vol. 35. National Acad Sciences, pp. 652–655.

[6] Feng, J., Lin, Z., Xu, H., Yan, S., 2014. Robust subspace segmentation with block-diagonal prior. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3818–3825.

[7] Georghiades, A. S., Belhumeur, P. N., Kriegman, D. J., 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6), 643–660.

[8] Hagen, L., Kahng, A. B., 1992. New spectral methods for ratio cut partitioning and clustering. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems 11 (9), 1074–1085.

[9] He, R., Zheng, W.-S., Hu, B.-G., Kong, X.-W., 2011. Nonnegative sparse coding for discriminative semi-supervised learning. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2849–2856.

[10] He, X., Cai, D., Yan, S., Zhang, H.-J., 2005. Neighborhood preserving embedding. In: IEEE International Conference on Computer Vision. Vol. 2. IEEE, pp. 1208–1213.

[11] He, X., Niyogi, P., 2004. Locality preserving projections. In: Advances in neural information processing systems. pp. 153–160.

[12] Huang, J., Nie, F., Huang, H., 2015. A new simplex sparse learning model to measure data similarity for clustering. In: International Joint Conference on Artificial Intelligence. pp. 3569–3575.

[13] Jain, A. K., 2010. Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31 (8), 651–666.

[14] Jia, H., Cheung, Y.-M., 2017. Subspace clustering of categorical and numerical data with an unknown number of clusters. IEEE transactions on neural networks and learning systems. DOI: 10.1109/TNNLS.2017.2728138.

[15] Jin, B., Vai, M. I., 2014. An adaptive ultrasonic backscattered signal processing technique for instantaneous characteristic frequency detection. Bio-medical materials and engineering 24 (6), 2761–2770.

[16] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., Wu, A. Y., 2002. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7), 881–892.

[17] Larsen, R. M., 2004. Propack-software for large and sparse svd calculations. Available online. URL http://sun.stanford. edu/rmunk/PROPACK, 2008–2009.

[18] Learned-Miller, E., Huang, G. B., RoyChowdhury, A., Li, H., Hua, G., 2016. Labeled faces in the wild: A survey. In: Advances in Face Detection and Facial Image Analysis. Springer, pp. 189–248.

[19] Li, S., Fu, Y., 2016. Learning robust and discriminative subspace with low-rank constraints. IEEE transactions on neural networks and learning systems 27 (11), 2160–2173.

[20] Lichman, M., 2013. Uci machine learning repository [http://archive.ics.uci.edu/ml].

[21] Lin, Z., Chen, M., Ma, Y., 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055.

[22] Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y., 2013. Robust recovery of subspace structures by low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1), 171–184.

[23] Liu, G., Yan, S., 2011. Latent low-rank representation for subspace segmentation and feature extraction. In: IEEE International Conference on Computer Vision. IEEE, pp. 1615–1622.

[24] Liu, J., Chen, Y., Zhang, J., Xu, Z., 2014. Enhancing low-rank subspace clustering by manifold regularization. IEEE Transactions on Image Processing 23 (9), 4022–4030.

[25] Liu, Q., Lu, X., He, Z., Zhang, C., Chen, W.-S., 2017. Deep convolutional neural networks for thermal infrared object tracking. Knowledge-Based Systems 134, 189–198.

[26] Lu, G.-F., Wang, Y., Zou, J., 2016. Low-rank matrix factorization with adaptive graph regularizer. IEEE Transactions on Image Processing 25 (5), 2196–2205.

[27] Lu, Y., Lai, Z., Li, X., Wong, W. K., Yuan, C., Zhang, D., 2018. Low-rank 2-d neighborhood preserving projection for enhanced robust image representation. IEEE Transactions on Cybernetics.

[28] Lu, Y., Yuan, C., Lai, Z., Li, X., Zhang, D., Wong, W. K., 2018. Horizontal and vertical nuclear norm-based 2dlda for image representation. IEEE Transactions on Circuits and Systems for Video Technology.

[29] Lu, Y., Yuan, C., Li, X., Lai, Z., Zhang, D., Shen, L., 2018. Structurally incoherent low-rank 2dlpp for image classification. IEEE Transactions on Circuits and Systems for Video Technology.

[30] Luo, G., Dong, S., Wang, K., Zuo, W., Cao, S., Zhang, H., 2017. Multi-views fusion cnn for left ventricular volumes estimation on cardiac mr images. IEEE Transactions on Biomedical Engineering.

[31] Ma, X., Liu, Q., Ou, W., Zhou, Q., 2018. Visual object tracking via coefficients constrained exclusive group lasso. Machine Vision and Applications, 1–15.

[32] Martinez, A. M., 1998. The ar face database. CVC technical report.

[33] Murtagh, F., 1983. A survey of recent advances in hierarchical clustering algorithms. The Computer Journal 26 (4), 354–359.

[34] Nene, S. A., Nayar, S. K., Murase, H., et al., 1996. Columbia object image library (coil-20).

[35] Ng, A. Y., Jordan, M. I., Weiss, Y., 2002. On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems. pp. 849–856.

[36] Nie, F., Wang, X., Huang, H., 2014. Clustering and projected clustering with adaptive neighbors. In: ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 977–986.

[37] Nie, F., Wang, X., Jordan, M. I., Huang, H., 2016. The constrained laplacian rank algorithm for graph-based clustering. In: AAAI Conference on Artificial Intelligence. pp. 1969–1976.

[38] Parimala, M., Lopez, D., Senthilkumar, N., 2011. A survey on density based clustering algorithms for mining large spatial databases. International Journal of Advanced Science and Technology 31 (1), 59–66.

[39] Phillips, J., Bruce, V., Soulie, F. F., 1999. Face recognition: From theory to applications.

[40] Ren, Y., Zhang, G., Yu, G., Li, X., 2012. Local and global structure preserving based feature selection. Neurocomputing 89, 147–157.

[41] Roweis, S. T., Saul, L. K., 2000. Nonlinear dimensionality reduction by locally linear embedding. science 290 (5500), 2323–2326.

[42] Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8), 888–905.

[43] Sun, F., Yao, Y., Chen, M., Li, X., Zhao, L., Meng, Y., Sun, Z., Zhang, T., Feng, D., 2017. Performance analysis of superheated steam injection for heavy oil recovery and modeling of wellbore heat efficiency. Energy 125, 795–804.

[44] Sun, F., Yao, Y., Li, X., 2018. The heat and mass transfer characteristics of superheated steam coupled with non-condensing gases in horizontal wells with multi-point injection technique. Energy 143, 995–1005.

[45] Sun, F., Yao, Y., Li, X., Yu, P., Ding, G., Zou, M., 2017. The flow and heat transfer characteristics of superheated steam in offshore wells and analysis of superheated steam performance. Computers & Chemical Engineering

100, 80–93.

[46] Toh, K.-C., Yun, S., 2010. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pacific Journal of Optimization 6 (615-640), 15.

[47] Von Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and computing 17 (4), 395–416.

[48] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y., 2010. Locality-constrained linear coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 3360–3367.

[49] Wang, S.-J., Yang, J., Sun, M.-F., Peng, X.-J., Sun, M.-M., Zhou, C.-G., 2012. Sparse tensor discriminant color space for face verification. IEEE Transactions on Neural Networks and Learning Systems 23 (6), 876–888.

[50] Wen, J., Han, N., Fang, X., Fei, L., Yan, K., Zhan, S., 2018. Low-rank preserving projection via graph regularized reconstruction. IEEE Transactions on Cybernetics.

[51] Wen, J., Xu, Y., Li, Z., Ma, Z., Xu, Y., 2018. Inter-class sparsity based discriminative least square regression. Neural Networks 102, 36–47.

[52] Wen, J., Zhang, B., Xu, Y., Yang, J., Han, N., 2018. Adaptive weighted nonnegative low-rank representation. Pattern Recognition 81, 326–340.

[53] Xu, R., Wunsch, D., 2005. Survey of clustering algorithms. IEEE Transactions on Neural Networks 16 (3), 645–678.

[54] Yi, S., He, Z., Cheung, Y.-m., Chen, W.-S., 2017. Unified sparse subspace learning via self-contained regression. IEEE Transactions on Circuits and Systems for Video Technology.

[55] Yi, S., Lai, Z., He, Z., Cheung, Y.-m., Liu, Y., 2017. Joint sparse principal component analysis. Pattern Recognition 61, 524–536.

[56] Yin, M., Gao, J., Lin, Z., 2016. Laplacian regularized low-rank representation and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (3), 504–517.

[57] Yu, K., Zhang, T., Gong, Y., 2009. Nonlinear learning using local coordinate coding. In: Advances in Neural Information Processing Systems. pp. 2223–2231.

[58] Zhang, Z., Lai, Z., Xu, Y., Shao, L., Wu, J., Xie, G.-S., 2017. Discriminative elastic-net regularized linear regression. IEEE Transactions on Image Processing 26 (3), 1466–1481.

[59] Zhang, Z., Liu, L., Shen, F., Shen, H. T., Shao, L., 2018. Binary multi-view clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[60] Zhang, Z., Xu, Y., Shao, L., Yang, J., 2018. Discriminative block-diagonal representation learning for image recognition. IEEE Transactions on Neural Networks and Learning Systems 29 (7), 3111–3125.

[61] Zhuang, L., Gao, H., Lin, Z., Ma, Y., Zhang, X., Yu, N., 2012. Non-negative low rank and sparse graph for semi-supervised learning. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2328–2335.

[62] Zou, H., Hastie, T., Tibshirani, R., 2006. Sparse principal component analysis. Journal of Computational and Graphical Statistics 15 (2), 265–286.