Pattern Recognition 81 (2018) 326–340

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Adaptive weighted nonnegative low-rank representation

Jie Wen[a,b], Bob Zhang[c,*], Yong Xu[a,b,d], Jian Yang[e], Na Han[f]

[a] Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, Guangdong, PR China
[b] Shenzhen Medical Biometrics Perception and Analysis Engineering Laboratory, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, Guangdong, PR China
[c] Department of Computer and Information Science, University of Macau, Taipa, Macau, PR China
[d] Key Laboratory of Network Oriented Intelligent Computation, Shenzhen 518055, Guangdong, PR China
[e] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, PR China
[f] School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, Guangdong, PR China

ARTICLE INFO

Article history:
Received 26 May 2017
Revised 12 December 2017
Accepted 4 April 2018
Available online 11 April 2018

Keywords:
Low-rank representation
Adaptive weighted matrix
Data clustering
Locality constraint

ABSTRACT

Conventional graph based clustering methods treat all features equally even if they are redundant features or noise in the stage of graph learning, which is obviously unreasonable. In this paper, we propose a novel graph learning method named adaptive weighted nonnegative low-rank representation (AWNLRR) for data clustering. Based on the observation that noise and outliers usually cannot be represented well and suffer from larger reconstruction errors than the important features (clean features) in low-rank or sparse representation, we impose an adaptive weighted matrix on the data reconstruction errors to reinforce the role of the important features in the joint representation and thus a robust graph can be obtained. In addition, a locality constraint, *i.e.*, distance regularization term, is introduced to capture the local structure of data and enable the obtained graph to be sparser. These appealing properties allow AWNLRR to well capture the intrinsic structure of data, and thus AWNLRR has potential to achieve a better clustering performance than other methods. Experimental results on synthetic and real databases show that the proposed method obtains the best clustering performance than some state-of-the-art methods.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Data clustering has gained a lot of attention in the fields of machine learning and data mining [1]. The main purpose of data clustering is to partition a collection of samples without any label information into respective groups such that samples in the same group are naturally a class. For this goal, many methods have been proposed in the past few years, such as the density based clustering [2], *K*-means clustering [3], hierarchical clustering [4], neural networks-based clustering [5], and spectral clustering [6], etc.

High-dimensionality is one of the most challenging problems in data clustering. Generally, high-dimensionality data usually contain large amounts of redundant features such as noise and outliers. These useless features may play the dominant role during the clustering process, which leads to a bad performance. Spectral clustering is one of the most effective clustering methods to address this issue by seeking a low-dimensional representation with powerful discriminability from the original high-dimensional data [6–8]. It usually uses a three-step approach to obtain the clustering result. Specifically, it first constructs a graph that reveals the similarity relationships among data, and then produces the low-dimensional representation based on the graph, followed by using *K*-means to partition the low-dimensional data into respective groups. Generally, the clustering performance is directly determined by the constructed graph. In other words, constructing a natural graph to capture the essential relationship of data is very important to the spectral clustering. Recent years, various graph learning approaches have been proposed by using different metrics to measure the essential relationships among samples. For example, Euclidean distance is widely used to construct the *knn*-graph for clustering [6,9]. *knn*-graph reveals the distribution relationships of samples in the Euclidean space. Based on the *knn*-graph, Roweis constructed a locality linear embedding graph (*LLE*-graph) to capture the representation relationships between sample and its nearest neighbors [10]. Both of *knn*-graph and *LLE*-graph use distance metric to capture the local geometric structure of data. The only difference between them is that elements in *knn*-graph represent the distance relationships of samples while in *LLE*-graph denote the representation ability or contribution in the joint linear representation. However,

* Corresponding author.
E-mail addresses: wenjie@hrbeu.edu.cn (J. Wen), bobzhang@umac.mo (B. Zhang), yongxu@ymail.com (Y. Xu), csjyang@mail.njust.edu.cn (J. Yang), hannagdut@126.com (N. Han).

https://doi.org/10.1016/j.patcog.2018.04.004
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

these distance based graph learning methods have the following two main issues: (1) they are sensitive to the selection of the nearest neighbor size; (2) they cannot find the real nearest neighbors for each sample when data are suffered from noise, so that the obtained graph cannot capture the intrinsic structure of data.

Recent years, the representation techniques such as sparse representation and low-rank representation have been witness a great development and attracted much attention in data clustering owing to their success in adaptively uncovering the intrinsic representation structures of data [11–14]. Based on the assumption that each data point can be efficiently represented by a linear combination of a few points from its own subspace, sparse subspace clustering (SSC) imposes the sparsity norm, *i.e.,* l_1 norm, to constrain the self-representation matrix so that a natural graph with adaptive nearest neighbors is achieved [15]. We refer to the graph obtained by SSC as l_1-graph. However, l_1-graph is constructed independently to each sample, thus the derived l_1-graph lacks the global information of data [12]. Compared with SSC, low-rank representation (LRR) jointly learns the graph by imposing the nuclear norm to constrain the self-representation matrix so that the global structure of data is captured [16,17]. We refer to the graph learned by LRR as l_*-graph. However, l_*-graph is often denser than the l_1-graph, which does not guarantee the locality. Besides, both of the l_*-graph and l_1-graph lack the physical interpretation to the similarity relationship of samples since they contain many negative elements. To overcome these issues and obtain a more reasonable graph, non-negative low rank and sparse graph (*NNLRS*-graph) learning method is proposed, in which the sparsity and nuclear norm are simultaneously imposed to constrain the nonnegative self-representation matrix [12]. Moreover, in order to simultaneously capture the local and global structures of data, various extensions of LRR have been proposed. For example, the Laplacian regularizer is imposed on the self-representation matrix to preserve the local structure that similar samples have similar representations [18,19]. A Gaussian function based weighted matrix is introduced to ensure that the dissimilarity samples have small representation coefficients and *vice versa* [20]. Based on the observation that the perfect graph with satisfactory performance should better have exactly block-diagonal structure, Feng et al. sought for such graph by introducing a novel graph Laplacian constraint into the SSC and LRR [21]. The biggest limitation of this method is that it needs to know the exact number of clusters of data in advance.

Although the above extension methods of LRR and SSC are proved to be effective under mixed conditions, they have a severe problem that all features are treated equally in the graph construction and data representation even if many features are redundant features or even noises. It should be pointed out that these redundant features and noise not only are useless, but also may be harmful to the representation. Especially when the percentage of those redundant features is larger than the useful features, the redundant features may play the dominant role in the self-representation. In this case, the learned graph is inaccurate and reveals the mendacious relationships of samples, which leads to a bad clustering performance. In this paper, we mainly propose a novel and simple approach to overcome this issue. We observe that the outliers or noises usually cannot be well represented. This observation is also proved in many references. For example, many references show that using the sparse norm and nuclear norm to model the noise has potential to detect them since they usually have large reconstruction errors in practice [16,22]. Inspired by this observation, we impose a weighted matrix on the data reconstruction errors so that the representation contributions of the important features will be improved and those of the useless features with large reconstruction errors will be reduced. This encourages us to learn a more robust graph to reveal the intrinsic similarity relationships of samples than other methods. Moreover, a locality constraint is introduced to capture the local intrinsic structure that nearest neighbors should have larger representation contributions. These meaningful factors enable the method to perform better than other methods. Experimental results show that the proposed method not only can learn a clearer graph and obtains the best clustering performance than other methods, but also is robust to noises. In summary, the proposed method has the following good properties.

(1) By integrating the local distance regularization term into LRR, the proposed method can simultaneously exploit both global and local structures of data, which ensures to learn a more reasonable graph.
(2) The nonnegative constraint not only greatly improves the interpretability of the graph, but also guarantees each sample to be in the convex hull of its nearest neighbors.
(3) By introducing an adaptive weighted matrix to regularize the data reconstruction errors, the representation contribution of the most important features will be improved while those of the redundant features will be reduced in the self-representation so that a more robust graph will be achieved.

The paper is organized as the following six sections. Section 2 briefly introduces some related works about representation based clustering and classification. Section 3 mainly presents the proposed graph learning model and its solution. In section 4, we analyze the proposed method from the aspect of computational complexity, convergence, and connections to other methods. Section 5 conducts several experiments to evaluate the proposed method. Section 6 offers the conclusion.

2. Related works

In this section we briefly introduce some related representation based clustering and classification methods. For convenience, we first introduce some notations used through the paper. Matrix. $X = [x_1, x_2, ..., x_n] \in R^{m\times n}$. is the original data, column $x_i (i \in [1, n])$ denotes the ith data point, m and n are the number of features and samples, respectively. $\|E\|_p$ is the l_p $(p = 1, \{2, 1\}, F)$ norm of matrix E, and some typical norm constraints are calculated as $\|E\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{n}|e_{ij}|$, $\|E\|_F = (\sum_{i=1}^{m}\sum_{j=1}^{n} e_{ij}^2)^{1/2}$, and $\|E\|_{2,1} = \sum_{j=1}^{n}(\sum_{i=1}^{m} e_{ij}^2)^{1/2}$, respectively, where e_{ij} denotes element of the ith row and jth column of matrix E. $\|Z\|_*$ is the nuclear norm of matrix Z and is calculated as the sum of all singular values of matrix Z. $\mathbf{1} \in R^{m\times n}$ is a matrix which all elements are 1, $\boldsymbol{1} \in R^{m\times 1}$ is a vector that all elements are 1.

2.1. Representation based subspace clustering

In this paper, we refer to the methods that learn a graph by using the representation techniques, such as sparse representation and low-rank representation, etc., as the representation based subspace clustering (RSC) method. RSC can be unified into the following general framework [15,16,23]:

$$\min_{Z,E} \Phi(Z) + \lambda\Psi(E) s.t. X = XZ + E \tag{1}$$

where E is the reconstruction errors. $\Psi(E)$ models different noises by using different norm constraints, such as $\|E\|_1$, $\|E\|_{2,1}$, and $\|E\|_F^2$. λ is the regularization parameter used to balance the importance of the corresponding term. $\Phi(Z)$ is the regularization functions with respect to variable Z. The purpose of model (1) is to learn the self-representation matrix $Z \in R^{n\times n}$ that can best uncover the intrinsic geometric structures reside in the high-dimensional data. For different RSC methods, the major difference is the choice of $\Phi(Z)$. For example, sparse subspace clustering (SSC) [15] constrains matrix Z with l_1 norm to capture the sparse representation

relationships among data. Low-rank representation (LRR) chooses the nuclear norm, *i.e.*, $\Phi(Z) = \|Z\|_*$, to capture the global representation structure of data for clustering [16,24].

Once obtaining the representation matrix Z, RSC methods obtain the final clustering result via the following three steps: (1) building an affinity matrix $W = (|Z| + |Z^T|)/2$; (2) producing the low-dimensional representations by performing the eigendecomposition on the Laplacian matrix of the affinity graph; (3) partitioning the derived low-dimensional representations into c clusters via K-means.

2.2. Representation based classification methods

Representation based classification (RC) methods are typical supervised classification methods which exploit label information during classification [25]. RC assumes that samples of the same class with the test sample contribute much more than those of other classes in the joint linear representation of using all samples to represent the test sample [26]. Based on this assumption, various RC methods have been proposed, in which sparse representation based classification (SRC) [27], collaborative representation based classification (CRC) [28], regularized robust coding (RRC) [29], and locality-constrained linear coding (LLC) [30], *etc.*, are the most well-known methods. These RC methods classify the test samples via the following three steps: (1) using all training samples to represent the test samples and calculating the corresponding representation vector; (2) producing the representation residual of each class with respect to the test sample; (3) classifying the test sample into the class with the minimum residual [31]. The representation residual in the second step can also be regarded as the representation contribution of the class in the joint linear representation. The minimum representation residual means the largest contribution of the corresponding class. For various RC methods, the major difference among them is the approach to learn the representation vector. In most cases, the objective function to learn the representation vector of these RC methods can be unified into the following model [32]:

$$\min_{\alpha} \|s \odot (y - X\alpha)\|_2^2 + \lambda\varphi(d \odot \alpha) \tag{2}$$

where $\odot$ denotes the element-wise multiplication, λ is the regularization parameter. $X = [x_1, x_2, \ldots, x_n] \in R^{m\times n}$ is the training set which contains all training samples, $y \in R^{m\times 1}$ is the test sample. $\varphi(d\odot\alpha)$ is a regularization term of α with different norm constraints, such as $\|d \odot \alpha\|_2^2$ and $\|d\odot\alpha\|_1$, etc. Vectors s and d are the prior knowledge which encourages the model to learn a more reasonable representation vector. The major differences among most of RC methods are the choices of parameters s and d, and the regularized norm of $\varphi(d\odot\alpha)$. For example, if $d = s = \mathbf{1}$ and $\varphi(d \odot \alpha) = \|\alpha\|_1$, then model (2) is degraded to the basic model of SRC. LLC uses the Gaussian distances between the test sample and training samples as prior knowledge d to avoid selecting training samples that are far from test sample y in the joint linear representation. By doing so, the representation contribution of those samples that are much more possible to be the same class with the test sample can be efficiently improved [30]. RRC imposes an adaptive weighted vector s derived from the reconstruction error to constrain the reconstruction term so that the representation contribution of the important features can be improved and the negative influence of redundant features or outliers can be eliminated to some extent [29]. Based on RRC, Zheng et al. proposed an iterative re-constrained group sparse classification (IRGSC) method which can adaptively learn a more flexible weight s to identify the outlier and inlier [32].

3. Adaptive weighted nonnegative low-rank representation

As introduced in previous section, graph learning is the most important step in unsupervised clustering. A good affinity graph that can best capture the intrinsic structures of data is the assurance to obtain a satisfactory performance. In this section, we mainly present a robust graph learning method, *i.e.*, adaptive weighted nonnegative low-rank representation (AWNLRR) for unsupervised clustering.

3.1. Motivations and model of AWNLRR

Both of representation based clustering and classification methods prove that the representation relationships among data contain much discriminant information. Thus capturing the representation structure of data is necessary for graph based clustering method. LRR and SSC are proved to be effective in capturing the global and local representation structures of data, respectively. However, these two methods treat all features equally in the linear representation, no matter whether they are outlier or not. This is harmful to capture the intrinsic representation structure of data. In real world applications, samples always have large dimensions and contain many redundant features and noise. A robust graph learning method should have the ability to identify the important features and reinforce the effect of them during graph learning so as to adaptively learn a more robust graph. Motivated by RRC [29] and IRGSC [32], we propose the following weighted nonnegative low-rank representation approach:

$$\begin{aligned} &\min_{Z,S} \left\|S^{1/2} \odot (X - XZ)\right\|_F^2 + \frac{\lambda_1}{2}\|S\|_F^2 + \lambda_2\|Z\|_* \\ &s.t. S \geq 0, S^T\mathbf{1} = \mathbf{1}, Z \geq 0 \end{aligned} \tag{3}$$

where Z is the affinity graph to be learned, S is the weighted matrix with positive values of all elements. $S^{1/2}$ is defined as an element-wise square root of S, *i.e.*, each element of $S^{1/2}$ is $\sqrt{s_{ij}}$. λ_1 and λ_2 are tunable parameters used to balance the importance of the corresponding terms. By imposing the weighted matrix to regularize the data reconstruction errors, the method will adaptively assign smaller weight to the feature with larger reconstruction error and assign larger weight to the important feature. The constraint term $S^T\mathbf{1} = \mathbf{1}$ ensures all samples to be treated equally. In addition, we can prove that optimizing the objective function (3) allows the proposed method to obtain a sparse weighted matrix.

Proposition 1. Suppose $E = X - XZ$ and elements of each column of E are not all 0, minimizing the optimization sub-problem to variable S, *i.e.*, $\min_{S\geq 0, S^T\mathbf{1}=\mathbf{1}} \|S^{1/2} \odot (X - XZ)\|_F^2 + \frac{\lambda_1}{2}\|S\|_F^2$, leads to a sparse weighted matrix.

Proof. Define $D = E \odot E$. Obviously, we have $D \geq 0$. Then problem $\min_{S\geq 0, S^T\mathbf{1}=\mathbf{1}} \|S^{1/2} \odot (X - XZ)\|_F^2 + \frac{\lambda_1}{2}\|S\|_F^2$ is equivalent to problem $\min_{S\geq 0, S^T\mathbf{1}=\mathbf{1}} \|S\|_F^2 + \frac{2}{\lambda_1}\|D \odot S\|_1$. It is also equivalent to the following n independent sub-problems $\min_{s_i\geq 0, {s_i}^T\mathbf{1}=1} \|s_i\|_2^2 + \frac{2}{\lambda_1}\|d_i \odot s_i\|_1$, $i = 1, \ldots, n$, where s_i and d_i are the ith column of matrices S and D, respectively. Problem $\min_{s_i\geq 0, {s_i}^T\mathbf{1}=1} \|s_i\|_2^2 + \frac{2}{\lambda_1}\|d_i \odot s_i\|_1$ can be viewed as a special case of Lasso problem [33], which will produce a sparse solution s_i. Specially, the sparse degree is controlled by the penalty parameter $2/\lambda_1$ [34]. Thus, we can conclude that solving problem (3) will produce a sparse weighted matrix S.

Most importantly, restricting the value of S in a reasonable range by using the regularization term $\frac{\lambda_1}{2}\|S\|_F^2$ and boundary constraints $S \geq 0$, $S^T\mathbf{1} = \mathbf{1}$ can avoid trivial solution to S [32,34]. Constraint $Z \geq 0$ ensures the learned graph to have good interpretabil-

ity for samples such that its each element directly reveals the intrinsic similar degree of the corresponding two samples. Moreover, the non-negativity constraint has potential to obtain a better performance in the representation based graph learning [35].

As introduced in the previous section, the local structure of data is useful and also reveals the intrinsic relationships of samples. To preserve the local structure, we further impose a distance regularization term to constrain the affinity matrix Z as follows

$$\begin{aligned}\min_{S,Z} \left\|S^{1/2}\odot(X-XZ)\right\|_F^2 &+ \frac{\lambda_1}{2}\|S\|_F^2+\lambda_2\|Z\|_* \\ &+\lambda_3\sum_{i,j=1}^{n}\left\|x_i-x_j\right\|_2^2 z_{ij} \\ s.t.\, S\geq 0, S^T\mathbf{1}=\mathbf{1}, Z\geq 0 &\end{aligned} \tag{4}$$

where λ_3 is the tunable regularization parameter. The third regularization term is used to preserve the local structure of data so that similar samples have similar representations [36].

Define the ith row and jth column element d_{ij} of matrix D is $d_{ij}=\|x_i-x_j\|_2^2$, then $\sum_{i,j=1}^{n}\|x_i-x_j\|_2^2 z_{ij}=Tr(D^TZ)$, where $Tr(\cdot)$ is the trace operation. Then model (4) is transformed into:

$$\begin{aligned}\min_{S,Z} \left\|S^{1/2}\odot(X-XZ)\right\|_F^2 &+ \frac{\lambda_1}{2}\|S\|_F^2+\lambda_2\|Z\|_*+\lambda_3 Tr\left(D^TZ\right) \\ s.t.\, S\geq 0, S^T\mathbf{1}=\mathbf{1}, Z\geq 0 &\end{aligned} \tag{5}$$

To avoid the negative influence that sample is selected to represent itself and the trivial solution that some samples are not selected in the joint linear representation, *i.e.*, some rows of Z are all zeros, we further constrain the affinity graph such that the values of its diagonal elements is zero and the sum of its each row is one. Then the final graph learning model of AWNLRR is written as follows:

$$\begin{aligned}&\min_{S,Z} \left\|S^{1/2}\odot(X-XZ)\right\|_F^2+\frac{\lambda_1}{2}\|S\|_F^2+\lambda_2\|Z\|_*+\lambda_3 Tr(D^TZ) \\ &\quad s.t.\, S\geq 0, S^T\mathbf{1}=\mathbf{1}, diag(Z)=0, Z\geq 0, Z\mathbf{1}=\mathbf{1}\end{aligned} \tag{6}$$

3.2. Solution to AWNLRR

There are two unknown variables need to be solved in an Eq. (6). Obviously, it is unrealistic to obtain its analytical solution. In this section, we use the alternating direction method of multipliers (ADMM) [37] to obtain the local optimal solution of variables S and Z. We first introduce two auxiliary variables E and U to make the optimization problem (6) separable as follows:

$$\begin{aligned}&\min_{S,Z,E,U} \left\|S^{1/2}\odot E\right\|_F^2+\frac{\lambda_1}{2}\|S\|_F^2+\lambda_2\|U\|_*+\lambda_3 Tr\left(D^TZ\right) \\ &\quad s.t.\, S\geq 0, S^T\mathbf{1}=\mathbf{1}, diag(Z)=0, Z\geq 0, Z\mathbf{1}=\mathbf{1}, \\ &\quad E=X-XZ, Z=U\end{aligned} \tag{7}$$

Compared with problem (6), the complexity of problem (7) seems to be increased. Fortunately, we can prove that it is still a two-block optimization problem which can be fast solved by the classical ADMM. Please refer to Section 4.2 for the detailed proof.

We first form the following augmented Lagrangian function

$$\begin{aligned}L(Z,S,E,U,C_1,C_2) = &\left\|S^{1/2}\odot E\right\|_F^2+\frac{\lambda_1}{2}\|S\|_F^2 \\ &+\lambda_2\|U\|_*+\lambda_3 Tr\left(D^TZ\right) \\ &+\frac{\mu}{2}\left(\left\|X-XZ-E+\frac{C_1}{\mu}\right\|_F^2+\left\|Z-U+\frac{C_2}{\mu}\right\|_F^2\right)\end{aligned} \tag{8}$$

where C_1 and C_2 are Lagrangian multipliers,μ is the penalty parameter. Then we can calculate each variable by fixing the remaining variables, respectively.

Step 1. Update S: By fixing variables Z, E, U, variable S can be calculated by minimizing the following problem:

$$\min_{S\geq 0, S^T\mathbf{1}=\mathbf{1}} \left\|S^{1/2}\odot E\right\|_F^2+\frac{\lambda_1}{2}\|S\|_F^2 \tag{9}$$

When E is fixed, Eq. (9) is equivalent to the following minimization problem:

$$\begin{aligned}&\min_{S\geq 0, S^T\mathbf{1}=\mathbf{1}} \sum_{i=1}^{m}\sum_{j=1}^{n}\left(s_{ij}e_{ij}^2+\frac{\lambda_1}{2}s_{ij}^2\right) \\ \Leftrightarrow &\min_{S\geq 0, S^T\mathbf{1}=\mathbf{1}} \sum_{i=1}^{m}\sum_{j=1}^{n}\left(s_{ij}+\frac{e_{ij}^2}{\lambda_1}\right)^2\end{aligned} \tag{10}$$

It is obvious that the problem (10) is independent for different j. So we can obtain S by solving its each column separately [34]. Each column s_j is calculated by solving the following minimization problem:

$$\min_{s_j>0, s_j^T\mathbf{1}=1} \sum_{j=1}^{n}\left\|s_j+\frac{1}{\lambda_1}f_j\right\|_2^2 \tag{11}$$

where f_j is the jth column of matrix $F=E\odot E$.

To calculate s_j, we first transform Eq. (11) into the following Lagrangian function

$$L\left(s_j,\eta,\beta_j\right)=\frac{1}{2}\left\|s_j+\frac{1}{\lambda_1}f_j\right\|_2^2-\eta_j\left(s_j^T\mathbf{1}-1\right)-\beta_j^Ts_j \tag{12}$$

where η_j and $\beta_j>0$ are the Lagrangian multipliers.

The optimal solution s_j can be obtained by setting the derivative of Eq. (12) with respect to s_j to zero:

$$\partial L\left(s_j,\eta_j,\beta_j\right)/\partial s_j=s_j+\frac{1}{\lambda_1}f_j-\eta_j\mathbf{1}-\beta_j=0 \tag{13}$$

According to the Karush–Kuhn–Tucker (KKT) condition that $\beta_j\odot s_j=0$ [34], we can obtain s_j:

$$s_j=\max\left(\eta_j\mathbf{1}-\frac{1}{\lambda_1}f_j,0\right) \tag{14}$$

According to the constraint $s_j^T\mathbf{1}=1$, we have

$$\begin{aligned}&\sum_{i=1}^{m}\left(\eta_j-\frac{1}{\lambda_1}f_{ij}\right)=1 \\ &\Rightarrow \eta_j=\frac{1}{m}+\frac{1}{m\lambda_1}\sum_{i=1}^{m}f_{ij}\end{aligned} \tag{15}$$

When η_j is calculated, we can obtain the optimal solution s_j by using (14) so that the optimal solution S is obtained.

Step 2. Update E: By fixing variables Z, S, U, we can obtain variable E by solving the following minimization problem

$$\min_{E}\left\|S^{1/2}\odot E\right\|_F^2+\frac{\mu}{2}\left\|X-XZ-E+\frac{C_1}{\mu}\right\|_F^2 \tag{16}$$

Define $G=X-XZ+\frac{C_1}{\mu}$, Eq. (16) can be rewritten as follows:

$$\begin{aligned}&\min_{E}\left\|S^{1/2}\odot E\right\|_F^2+\frac{\mu}{2}\|E-G\|_F^2 \\ \Leftrightarrow &\min_{E}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(s_{ij}e_{ij}^2+\frac{\mu}{2}\left(e_{ij}-g_{ij}\right)^2\right) \\ \Leftrightarrow &\sum_{i=1}^{m}\sum_{j=1}^{n}\min_{e_{ij}}\left(e_{ij}-\frac{\mu g_{ij}}{\mu+2s_{ij}}\right)^2\end{aligned} \tag{17}$$

From problem (17), we can obtain that the optimal solution to each element e_{ij} of variable E is:

$$e_{ij}=\frac{\mu g_{ij}}{\mu+2s_{ij}} \tag{18}$$

So that variable E is obtained. Theoretically, variables S and E need to be iteratively updated in a sub-loop. In this paper, we only update the two variables one time in a loop for computational efficiency.

Step 3. Update U. Fix variables Z, S, E, variable U can be obtained by minimizing objective function L with respect to U as follows:

$$\min_U \lambda_2\|U\|_* + \frac{\mu}{2}\left\|Z - U + \frac{C_2}{\mu}\right\|_F^2 \tag{19}$$

Problem (19) has a closed solution as follows:

$$U = \Theta_{\lambda_2/\mu}(Z + C_2/\mu) \tag{20}$$

where Θ is the singular value thresholding (SVT) shrinkage operation [16].

Step 4. Update Z: By fixing variables S, E, U, we can calculate variable Z by minimizing the following equation:

$$\min_Z \lambda_3 Tr\left(D^T Z\right) + \frac{\mu}{2}\left\|Z - U + \frac{C_2}{\mu}\right\|_F^2 + \frac{\mu}{2}\left\|X - XZ - E + \frac{C_1}{\mu}\right\|_F^2 \quad s.t. diag(Z) = 0, Z \geq 0, Z\mathbf{1} = \mathbf{1} \tag{21}$$

For simplicity and computational efficiency, we first calculate a latent solution $\hat{Z}$ by minimizing the following problem

$$\hat{Z} = \arg\min_Z \lambda_3 Tr\left(D^T Z\right) + \frac{\mu}{2}\left\|Z - U + \frac{C_2}{\mu}\right\|_F^2 + \frac{\mu}{2}\left\|X - XZ - E + \frac{C_1}{\mu}\right\|_F^2 \tag{22}$$

Problem (22) has a closed solution as

$$\hat{Z} = \left(X^T X + I\right)^{-1}\left(X^T M_1 + M_2 - \frac{\lambda_3}{\mu} D\right) \tag{23}$$

where $M_1 = X - E + \frac{C_1}{\mu}$, $M_2 = U - \frac{C_2}{\mu}$.

Then the optimal solution Z can be calculated by solving the following minimization problem:

$$\min_{diag(Z)=0, Z\geq 0, Z\mathbf{1}=\mathbf{1}} \left\|Z - \hat{Z}\right\|_F^2 \tag{24}$$

Similar to the optimization style of problem (10), we can obtain the optimal solution Z. For each row of Z, its optimal solution is

$$z_i = \max\left(\zeta_i \bar{\mathbf{1}}^T + \bar{z}_i, 0\right) \tag{25}$$

where $\bar{z}_i = [\bar{z}_{i1}, \ldots, \bar{z}_{ii}, \ldots, \bar{z}_{in}]$ is the ith row of $\hat{Z}$ (obtained by Eq. (23)) that element $\bar{z}_{ii}$ is set to zero. $\bar{\mathbf{1}}$ is the column vector that all elements except the ith element are one and the ith element is zero. ζ_i is the Lagrangian multiplier that is calculated as:

$$\xi_i = (1 + \bar{z}_i \mathbf{1})/(n-1) \tag{26}$$

For each row, after computing ξ_i, we can obtain the optimal solution of z_i by Eq. (25) so that the optimal solution Z is obtained.

Step 5. Update C_1, C_2 and μ. Lagrangian multipliers C_1 and C_2, penalty parameter μ are updated as follows:

$$C_1 = C_1 + \mu(X - XZ - E) \tag{27}$$

$$C_2 = C_2 + \mu(Z - U) \tag{28}$$

$$\mu = \begin{cases} \min(\mu_{\max}, \rho\mu), & if\ \pi < 0.01 \\ \mu, & else \end{cases} \tag{29}$$

where parameters $\mu_{\max}$ and ρ are positive constants, $\pi = \max(\|Z_k - Z_{k-1}\|_F, \|U_k - U_{k-1}\|_F, \|E_k - E_{k-1}\|_F)/\|X\|_F$, Z_k, U_k, E_k and $Z_{k-1}, U_{k-1}, E_{k-1}$ are the value of Z, U, E at the kth iteration (current step) and $k-1$th iteration (previous step), respectively.

The optimization steps of AWNLRR are summarized in Algorithm 1.

Algorithm 1 AWNLRR (solving (7)).

Input: Data matrix X, Parameters λ_1, λ_2, λ_3
Initialization: Constructing the k-nearest neighbor graph as the initial matrix of Z; $S = \mathbf{1}$, $U = Z$, $E = X - XZ$; $C_1 = C_2 = 0$, $\mu = 0.01$, $\rho = 1.1$, $\mu_{\max} = 10^8$.
while not converged **do**
1. Update S by using Eq. (14).
2. Update E by using Eq. (18).
3. Update U by using Eq. (20).
4. Update Z by using Eq. (25).
5. Update C_1, C_2, μ by using Eqs. (27), (28), and (29), respectively.
end while
Output: Z, S

Algorithm 2 Data clustering via Ncut with the obtained graph Z.

Input: Graph Z obtained via Algorithm 1, cluster number c
1. Calculate the affinity matrix $W = (|Z| + |Z^T|)/2$.
2. Compute the normalized Laplacian matrix $L = I - D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix with each diagonal element $d_{ii} = \sum_{j=1}^{n} w_{ij}$, I is the identity matrix.
3. Perform eigenvalue decomposition on matrix L and treat the first c eigenvectors corresponding to the first c smallest eigenvalues as the new representation $Y \in R^{n\times c}$ of original data, where each row vector y_i can be regarded as the new representation of the ith original sample.
4. Obtain the normalized new representation of each sample by $\hat{y}_i = y_i/\|y_i\|_2$.
5. Obtain c clusters $F_1, \ldots, F_c$ by performing K-means on the normalized new representations.
Output: Clusters $A_1, \ldots, A_c$ with $A_i = \{j | y_j \in F_i\}$.

3.3. AWNLRR based clustering

Similar to conventional graph based clustering methods, we also use the spectral clustering to obtain the final clustering results. Normalized cut (Ncut) [9] and Ratio cut (Rcut) [38] are the two most popular spectral clustering algorithms. The only difference between them is that Rcut produces the low-dimensional representation from the conventional Laplacian matrix derived from the similarity graph Z, while Ncut produces the low-dimensional representation from the normalized Laplacian matrix. In this paper, we chose Ncut to partition data into respective groups when similarity graph Z is obtained. The clustering steps via Ncut are summarized in Algorithm 2 in details.

3.4. Out-of-sample extension

The graph based clustering can only partition the available data that used during graph learning into respective groups. They cannot deal with new sample that does not used to learn the graph. Generally, there are two approaches which are widely applied to address this problem when the graph is obtained. The one approach first produces a linear projection from the obtained graph, and then classifies the new sample in the low-dimensional subspace [39]. This approach can also be viewed as the graph based dimensionality reduction. The second approach uses the well-known supervised classification method, *i.e.*, representation based classification, such as collaborative representation based classification (CRC) and sparse representation based classification (SRC), to classify the new sample [40,41]. Considering that the first approach is sensitive to the selection of dimension, in this paper we adopt the second approach to address the out-of-sample problem. In particular, we chose the CRC [42] to classify the new sample owing to its good performance and high efficiency. The detailed steps to address the out-of-sample problem via CRC are summarized in Algorithm 3.

Algorithm 3 New sample classification based on AWNLRR.

Input: Training data $X \in R^{m\times n}$, test sample $y \in R^{m\times 1}$, parameter β.

1. Use Algorithm 2 to obtain the clustering results of training data X;
2. Represent the test sample y by the linear combination of all samples in the training data X, and calculate the representation vector $\alpha = \arg\min_{\alpha} \|y - X\alpha\|_F^2 + \frac{\beta}{2}\|\alpha\|_2^2$;
3. Calculate the normalized representation residual of each cluster by $r_i = \|y - X_i\alpha_i\|_2 / \|\alpha_i\|_2$, where X_i denotes the training samples from the ith cluster and α_i is their corresponding representation coefficients;
4. Classify the test sample to the cluster with the minimum representation residual as $identity(y) = \arg\min_i r_i$.

Output: Predict label $identity(y)$.

4. Analysis of the proposed method

4.1. Computational complexity of AWNLRR

For AWNLRR listed in Algorithm 1, there are five main steps, in which the most computational costs are the SVT and inverse operation of matrix in steps 3 and 4, respectively. It should be noted that steps 1 and 2 can be viewed as the element-wise operation which can be fast solved. This indicates that the computational complexities of these two steps are very low and thus can be ignored compared with the other steps. Besides, we do not take into account the computational complexity of fundamental matrix operations, such as matrix addition and multiplication. For step 4, although the matrix inverse operation $(X^TX + I)^{-1}$ has high computational complexity, we can pre-calculate it before iteration loop because the matrix inverse operation is independent with all variables. So the real computational cost of step 4 is only the matrix multiplication operation which can be ignored. For a matrix U with the size of $n \times n$, the computational complexity of SVT operation is $O(rn^2)$ by using the skinny singular value decomposition (SVD), where r ($r \le n$) is the rank of matrix U [18]. So the computational complexity of step 3 is about $O(rn^2)$. Therefore, the total computational complexity of the proposed method is about $O(\tau rn^2)$, where τ is the iteration number.

4.2. Convergence analysis of AWNLRR

As presented in previous section, the ADMM is adopted to solve the optimization problem (6). In this subsection, we mainly focus on analyzing the convergence property of the proposed method with the proposed optimization scheme listed in Algorithm 1.

Proposition 2. The optimization problem (7) is equivalent to the two-block optimization problem. And the proposed Algorithm 1 is equivalent to the classical ADMM for the two-block problem.

Proof. The classical two-block optimization problem can be unified as follows [43,44]:

$$\min_{W\in\Omega_W, Y\in\Omega_Y} f(W) + g(Y) s.t. AW + BY = L \tag{30}$$

where Ω_W and Ω_Y are the domains (boundary constraints) of variables W and Y. $f(\cdot)$ and $g(\cdot)$ are the convex functions. A, B, L could be either vectors or matrices. Classical ADMM first converts the constrained optimization problem (30) into the following augmented Lagrangian function:

$$L(W, Y, C) = f(W) + g(Y) + \frac{\mu}{2}\left\|AW + BY - L + \frac{C}{\mu}\right\|_F^2 \tag{31}$$

Then iteratively update all variables as follows

$$W_{t+1} = \arg\min_{W\in\Omega_W} L(W, Y_t, C_t) \tag{32}$$

$$Y_{t+1} = \arg\min_{Y\in\Omega_Y} L(W_{t+1}, Y, C_t) \tag{33}$$

$$C_{t+1} = C_t + \mu(AW_{t+1} + BY_{t+1} - L) \tag{34}$$

From Algorithm 1, it is obvious to see that our optimization problem (7) is optimized by the similar approach with the classical ADMM [45]. Specially, the optimization of variable Z in (21) is equivalent to optimize Y in (33) when other variables are fixed. For variable U, we can find that the optimization of U is independent with variables S and E. While optimizing variables S and E can be treated as a unified sub-problem during the optimization, which can be calculated by the block coordinate-wise descent method. It should be noted that we only update variables S and E one time for computational efficiency in Algorithm 1. In this case, the optimization steps for variables S, E, U can be accumulated in W as (32) [45]. Hence, the optimization problem (7) can be viewed as a special case of the classical two-block optimization problem. And the proposed optimization algorithm, *i.e.*, Algorithm 1, is equivalent to the classical ADMM for the two-block problem. Thus we complete the proof.

For the classical two-block ADMM, the convergence property has been theoretically proved in [43,46–48]. Hence, as an equivalent two-block optimization problem, the proposed optimization approach can also converge to the local optimum as the classical ADMM.

We also conduct experiments to prove the convergence property of the proposed algorithm. Fig. 1 shows the objective function value and clustering accuracy (%) *versus* the iteration step, in which the objective function value is calculated as $obj = \|S^{1/2} \odot E\|_F^2 + \frac{\lambda_1}{2}\|S\|_F^2 + \lambda_2\|U\|_* + \lambda_3 Tr(D^TZ) + \|X - XZ - E\|_F^2 + \|Z - U\|_F^2$. It is obvious that the objective function value is monotonically decreasing till to the stable point, which also proves the fast convergence property of the proposed method.

4.3. Connections to other methods

Since the proposed method utilizes the low-rank representation technique to capture the structure of data, thus we mainly analyze the connections between the method and other LRR based graph learning methods, such as LRR [16], Laplacian regularized LRR (LapLRR) [19], NNLRS [12], and non-negative sparse hyper-Laplacian regularized LRR (NSHLRR) [18], etc.

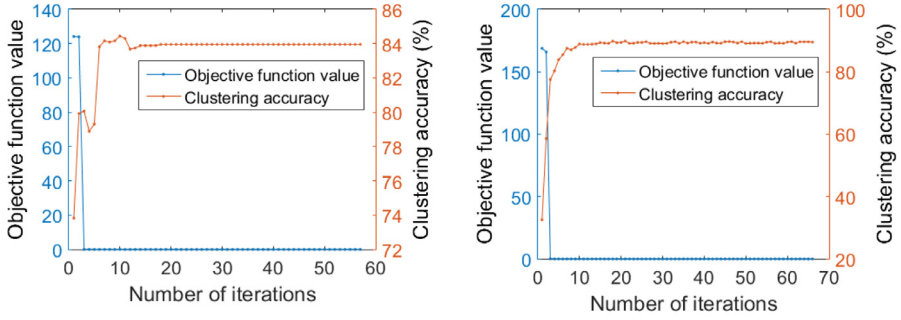
(1) Connections to LRR and NNLRS: The graph learning model of LRR is briefly introduced in Section 2.1. NNLRS is an extension of LRR. It seeks a non-negative graph that can capture both global and local structures of data. The graph learning model of NNLRS is as follows

$$\min_{Z,E} \|Z\|_* + \beta\|Z\|_1 + \lambda\|E\|_{2,1} s.t. X = XZ + E, Z \ge 0 \tag{35}$$

By introducing the sparse constraint $\beta\|Z\|_1$, NNLRS has potential to learn a sparser graph than LRR. Parameter β controls the sparse degree of the learned graph.

It should be noted that LRR and NNLRS can be viewed the special cases of AWNLRR. When all elements of S are defined as $s_{ij} = 1/m$ (m is the feature dimension of matrix X), $\lambda_1 = \lambda_3 = 0$, AWNLRR degrades into a variation of LRR, in which the only difference between them is the regularization of error term. If $\lambda_1 = 0$, D and S are given as follows: all elements of D are equivalent, and all elements of S are $1/m$, then AWNLRR degrades into a variation of NNLRS to some extent, in which the only difference is also the regularization of error term.

Compared with LRR and NNLRS, the proposed method has the following advantages. First, both of LRR and NNLRS only capture the representation structures of data while ignoring the distance or nearest neighbor relationships of samples. Compared with these



(a) COIL20 (b) Extended Yale B

Fig. 1. Objective function value and clustering accuracy *versus* the number of iteration of the proposed method on the COIL20 and Extended Yale B databases.

two methods, AWNLRR utilizes a weighted sparse constraint,[1] *i.e.*, $\lambda_3 Tr(D^T Z)$, to capture the local structure of data. It not only ensures the sparsity, but also effectively exploits the local distance information of samples to learn a more reasonable graph. Second, during the data representation of LRR and NNLRS, all features no matter they are redundant features or noise are treated equally. This is harmful to obtain the clean graph. The proposed method effectively tackles this issue by introducing an adaptive weighted matrix S. By boosting the two variables of S and Z together, AWNLRR has potential to improve the role of those most important features during data representation, so that a more robust graph will be produced.

From the above analyses, AWNLRR has many superior properties compared with these two methods, which enables it to obtain a better performance.

(2) Connections to LapLRR and NSHLRR: The graph learning models of LapLRR and NSHLRR are shown as Eq. (36) and Eq. (37), respectively.

$$\min_Z \frac{1}{2}\|X - XZ\|_F^2 + \lambda_1 \|Z\|_* + \frac{\lambda_2}{2} Tr\left(ZLZ^T\right) s.t. Z \geq 0 \tag{36}$$

$$\min_Z \|Z\|_* + \lambda_1 \|Z\|_1 + \beta Tr\left(ZL^h Z^T\right) + \gamma \|X - XZ\|_1 s.t. Z \geq 0 \tag{37}$$

where L and L^h are the Laplacian graphs and Laplacian hypergraph [18].

Compared with LapLRR, NSHLRR additionally introduces a sparsity term, *i.e.*, $\lambda_1 \|Z\|_1$, to capture the local representation structure of data and avoid a dense graph. Although nearest neighbor relationships are exploited in these two methods, they cannot ensure the greater contributions of nearest neighbors during data representation. Compared with these two methods, AWNLRR simply imposes a simple distance regularization term rather than Laplacian regularization to constrain the graph. In this way, the representation coefficients of these nearest neighbors will be enlarged such that the representation contributions of them will also be improved. Moreover, the distance regularization term also ensures the sparsity of the graph. These properties are beneficial to obtain a more reasonable and interpretable graph that each element naturally reveals the similarity degree of the corresponding two samples. Similar to the previous analysis, the adaptive weighted matrix encourages the method to obtain a more robust graph than these two methods. In summary, AWNLRR has potential to learn a more reasonable, interpretable, and robust graph than LapLRR and NSHLRR.

[1] If $Z \geq 0$ and $D > 0$, $Tr(D^T Z) = |D \odot Z|_1$. Therefore term $Tr(D^T Z)$ can be regarded as the weighted sparse term.

5. Experiments and analysis

In this section, several experiments are conducted on both synthetic and real databases to evaluate the clustering performance of AWNLRR. *K*-means and several graph based clustering methods, including ratio cut (Rcut) [38], Normalized cut (Ncut)[2] [9], SSC[3][15], LRR[4] [16], Latent LRR (LatLRR) [49], LapLRR [19], NSHLRR[5] [18], and principal graph and structure learning (PGSL)[6] [50], are chose to compare the proposed method to prove its effectiveness. The compared Rcut and Ncut methods use the *knn*-graph to perform clustering. LatLRR is an extension of LRR which learns a graph by effectively exploiting the hidden data. Based on reversed graph embedding, PGSL learns a principle graph that captures the local information for data clustering. Two metrics, *i.e.*, clustering accuracy (Acc) and normalized mutual information (NMI) [51] are chose as the evaluation criterion to compare different clustering methods. All experiments are performed on the same platform, *i.e.*, software Matlab 2015b and Windows 10 system, hardware Intel Core i7-4790 CPU and 16GB ram. In this work, parameters of all compared methods are manually tuned in a wide range to obtain their best results. Moreover, since *K*-means is sensitive to the initialization, thus we run these methods 15 times and then report their mean values for comparing. In the following experiments, the nearest neighbor size of the initial graph Z of the proposed method is set as 10.

5.1. Experiments on synthetic data

In this section, we compare different methods on the synthetic database. We use the method presented in [19,52] to generate a data matrix $X \in R^{300 \times 200}$ with 5 clusters. Each cluster contains 40 samples and each sample has 300 features. Then we further add 'salt and pepper' noise with different densities on the ground truth data matrix X to produce some noisy data to validate the robustness of these clustering methods.

Experimental results of different clustering methods on these noisy data are shown in Table 1. It should be pointed out that the data with noisy density of 0.0 is the original clean data X. From Table 1, it is obvious to see that all clustering methods achieve 100% accuracy and NMI on the original clean synthetic data. While with the increasing of the noisy density, their performances decrease dramatically, especially Rcut and Ncut. This indicates that the distance metric cannot correctly uncover the intrinsic near-

[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
[6] Code of PGSL is available at: http://liwang8.people.uic.edu/#publication

Table 1
Clustering accuracies (%) of different methods on the synthetic databases with different noisy degrees.

Metrics	Density	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	LapLRR	NSHLRR	PGSL	AWNLRR
Accuracy	0.0	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	0.1	84.75	52.50	55.77	**100**	84.50	**100**	75.65	**100**	63.60	**100**
	0.2	36.95	35.47	32.60	97.00	37.30	97.50	35.40	**99.50**	32.45	99.00
	0.3	28.95	31.50	32.00	73.50	30.65	88.50	32.50	86.00	30.25	**94.45**
	0.4	28.65	32.50	33.27	63.50	28.00	84.00	29.40	68.00	29.20	**92.50**
	0.5	28.25	32.73	29.30	30.55	29.00	31.00	30.10	37.00	29.40	**41.00**
NMI	0.0	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	0.1	65.09	24.86	29.88	**100**	62.99	**100**	50.98	**100**	42.29	**100**
	0.2	10.96	8.03	6.86	91.69	10.82	93.13	9.49	**98.54**	6.65	97.09
	0.3	3.96	4.35	6.70	46.14	6.55	70.28	7.20	67.00	4.20	**82.63**
	0.4	4.08	6.98	7.24	30.31	4.23	63.88	4.50	47.39	3.87	**67.46**
	0.5	3.38	5.47	4.64	5.00	4.13	5.06	7.44	8.47	4.67	**15.73**

Note: bold numbers denote the best results.

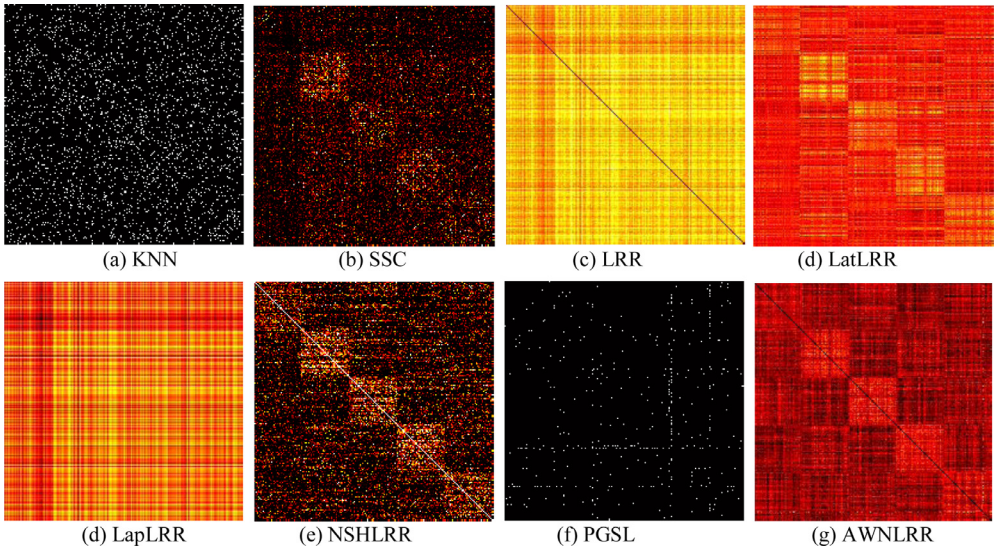


Fig. 2. Graph obtained by different graph learning methods on the synthetic database with noisy density of 0.3. Note: for *knn*-graph, the nearest neighbor size is set as 10. All graphs are showed with 'hot' colormap.

est neighbor structure of data when data contain dense noise. Yet, the proposed method achieves the better performance than other methods in almost all cases. In particular, when the noisy densities are 0.3 and 0.4, the proposed method still achieves the outstanding accuracies and NMIs, whose accuracies are about 6% higher than the second best method, *i.e.*, LatLRR. These outstanding performances indicate that the proposed method has potential to learn a more robust graph than other methods when data is corrupted with noise.

Fig. 2 shows some graphs obtained by different compared methods on the synthetic database with noisy density of 0.3. It is obvious to see that graphs obtained by SSC, LatLRR, NSHLRR, and AWNLRR has clearer block structure than those of KNN, LRR, LapLRR, and PGSL. In view of SSC, NSHLRR, and AWNLRR all impose the sparse constraint on the graph, we can conclude that the sparse representation is more effective than the low-rank representation in uncovering the intrinsic structures of noisy data. Although graph learned by NSHLRR is sparser than those of LatLRR and AWNLRR, its clustering accuracy and NMI are lower than those of LatLRR and AWNLRR. From Fig. 2(d), (e) and (g), one can see that the block diagonal structure of graphs learned by LatLRR and AWNLRR are clearer than that of NSHLRR. This indicates that the clearer block structure the constructed graph, the better the clustering performance. Compared with NSHLRR which shares some similar properties with the proposed method, the better performance of AWNLRR indicates that the extra non-negative weighted constraint of the proposed method is useful and effective to identify those important features (clean features) and reinforce their roles in graph learning, which has potential to learn a more robust graph from the noisy data.

5.2. Experiments on real datasets

In this section, we conduct experiments on some real databases listed in Table 2, including handwritten digit databases, face databases, object databases, and some non-image databases from University of California, Irvine (UCI) [53].

Handwritten digit database: USPS[7] and MNIST[8] are the two most well-known handwritten digit databases. They contain 10 classes from digit of "0″ to "9″. In the experiments, two subsets of these two databases which respectively contain 4000 and 6996 Gy digit images are selected for comparing. Typical images of these two datasets are shown in Fig. 3. Sizes of each image in USPS and MNIST are 16×16 and 28×28, respectively.

Face databases: The above methods are compared on five typical face databases, *i.e.*, the Umist face database[9] [56], the Extended Yale B (YaleB) face database [54], the AR face database [55], the Labeled Faces in the Wild (LFW) face dataset [58], and the MSRA

[7] Available at: http://www.gaussianprocess.org/gpml/data/
[8] Available at: http://yann.lecun.com/exdb/mnist/
[9] Available at: http://cs.nyu.edu/~roweis/data.html

Table 2
Description of databases.

	Database	No. of instances	Dimensions	Classes
Handwritten digit databases	USPS	4000	256	10
	MNIST	6996	784	10
Face databases	YaleB [54]	2414	1024	38
	AR [55]	3120	2000	120
	Umist [56]	575	2576	20
	LFW [58]	1251	1024	86
	MSRA	1799	256	12
Object databases	COIL20 [57]	1440	1024	20
UCI databases	Cars [53]	392	8	3
	Vehicle [53]	846	18	4
	Yeast [53]	1484	8	10

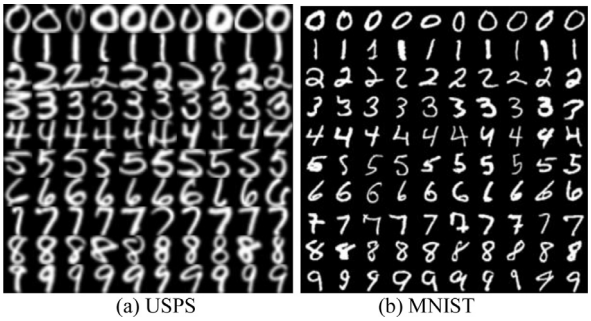


Fig. 3. Typical samples of the (a) USPS database and (b) MNIST database.

Fig. 4. Typical images of the used face databases, in which images from the first row to the last row are from the Umist, YaleB, AR, LFW, and MSRA databases, respectively.

database.[10] The Umist face database used in this work has 575 images provided by 20 persons under different poses. The used YaleB face database has 38 classes and 2414 images acquired from different illumination sceneries. The used AR face database contains 120 persons and 3120 images in total, in which each person provides 26 images with different facial expressions, illumination conditions, and occlusions by sun glasses and scarf. There are more than 13,000 images in the original LFW database which are collected from the web. In this work, a subset which contains 1251 face images of 86 persons is adopted for evaluation. The MSRA database contains 12 persons and 1799 images in total. Typical images of these databases are shown in Fig. 4. Sizes of images of the above face databases used in the experiments are 32×32, 50×40, 56×46, 32×32, and 16×16, respectively.

Fig. 5. Typical images of the COIL20 database.

Object database: In this work, we choose the Columbia Object Image Library (COIL20) database[11] [57] as the representation of the object database to evaluate those compared methods. The COIL20 database contains 1440 gray-scale images provided by 20 objects. There are 72 images of each object which are taken at pose intervals of 5°. Images used in this work were pre-resized to 32×32 for computational efficiency.

Non-image databases from UCI: The UCI Machine Learning Repository collects lots of databases. In this work, we select many non-image databases, including the Cars, Vehicle, Isolet, and Yeast databases[12] for clustering evaluation.

Specially, we deeply compare the above clustering methods on the USPS, COIL20, YaleB, and Umist datasets, in which a series of experiments are conducted on a range of first c sub-classes of these databases. For the remaining databases, we directly perform those methods on the corresponding whole database for evaluation (Fig. 5).

Experimental results of different clustering methods on the above databases are shown in Table 3–7 and Fig. 6. From these tables and figures, one can obtain that:

(1) From the comparison of K-means and other graph based clustering methods, it is obvious to see that learning a low-dimensional representation is effective to obtain a better clustering performance than using the original features directly. Most importantly, from Table 3–7 and Fig. 6, we can find that the proposed method obtains the best performance in almost all cases.
(2) Table 3 and Fig. 6(a) show the clustering accuracies and NMIs of different methods on the object database, *i.e.*, COIL20 database. One can see that in most cases, the clustering accuracies and NMIs of Rcut and Ncut are much higher than the representation based graph learning methods, *i.e.*, SSC, LRR, and LatLRR, etc. This demonstrates that the distance metric captures the structure of data more accurately than the representation based metrics on the COIL20 database.
(3) From the comparison of SSC, LRR, LatLRR, LapLRR, and NSHLRR in Table 3–7 and Fig. 6, we can conclude that a better performance can be obtained by integrating the manifold information, *i.e.*, nearest neighbor relationships. It should be noted that the nearest neighbor relationships are also the distance relationships between sample and its nearest neighbors. Therefore, this also proves the valuable of distance relationships of samples to

[10] Available at: http://www.escience.cn/people/fpnie/papers.html

[11] Available at: http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[12] Available at: http://www.escience.cn/people/fpnie/papers.html

Table 3
Clustering Acc (%) of different methods on the COIL20 datasets.

No. of class	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	LapLRR	NSHLRR	PGSL	AWNLRR
4	62.15	86.46	82.64	62.50	96.53	91.32	96.53	98.37	81.25	**100**
6	48.36	91.68	92.05	62.70	64.12	67.59	81.90	85.48	76.27	**93.38**
8	43.66	86.88	86.96	77.26	70.83	65.28	74.31	78.24	78.07	**92.08**
10	46.06	83.82	86.04	67.11	68.47	68.22	75.01	80.17	77.64	**86.59**
12	53.41	83.16	81.70	79.98	62.99	66.81	78.54	83.65	81.01	**88.66**
14	56.81	83.01	81.97	74.01	66.36	75.00	78.98	80.24	78.40	**90.02**
16	60.83	82.35	79.81	75.28	69.33	71.25	77.95	81.87	75.95	**92.10**
18	64.96	81.09	82.62	75.53	66.54	67.18	80.94	83.74	82.89	**92.90**
20	57.67	76.49	77.83	77.92	66.39	65.64	75.01	81.48	80.76	**84.03**

Note: bold numbers denote the best results.

Table 4
Clustering Acc (%) of different methods on the YaleB datasets.

No. of class	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	LapLRR	NSHLRR	PGSL	AWNLRR
2	50.85	94.53	94.53	**100**	78.13	96.88	99.22	99.22	98.44	99.22
8	18.91	50.03	50.30	**88.31**	83.79	83.20	83.67	84.05	62.83	84.47
14	15.95	54.33	54.60	78.71	**89.46**	82.06	77.64	83.24	54.79	88.78
20	12.11	55.09	55.89	76.84	**90.44**	80.10	76.55	86.78	52.96	90.36
26	11.35	56.14	56.70	76.89	87.39	74.79	75.10	80.47	49.67	**92.50**
32	10.76	51.44	51.46	76.12	80.65	77.18	81.23	83.96	45.87	**91.92**
38	9.39	48.77	49.42	73.89	70.34	78.88	77.29	80.54	42.89	**88.89**

Note: bold numbers denote the best results.

Table 5
Clustering Acc (%) of different methods on the Umist datasets.

No. of class	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	LapLRR	NSHLRR	PGSL	AWNLRR
4	47.97	66.34	67.05	60.16	61.79	51.55	84.55	88.62	78.78	**93.50**
6	52.91	73.37	73.57	70.35	68.02	50.12	82.56	88.90	84.71	**90.70**
8	48.45	74.70	76.90	67.42	70.47	60.37	86.85	80.69	71.03	**97.18**
10	44.57	68.63	68.58	71.70	74.60	70.04	77.64	78.53	69.43	**81.43**
12	44.66	69.27	69.37	67.00	64.87	68.77	69.78	72.22	64.63	**79.04**
14	41.52	69.85	69.46	71.67	60.18	69.56	73.56	79.35	64.01	**82.24**
16	39.84	60.57	61.37	65.97	54.63	64.75	65.74	**71.35**	61.61	70.65
18	38.78	61.34	62.00	67.30	55.90	61.34	65.40	68.73	63.59	**69.76**
20	41.58	62.33	62.57	63.48	56.28	61.04	65.78	66.15	65.67	**70.10**

Note: bold numbers denote the best results.

Table 6
Clustering Acc (%) of different methods on the USPS datasets.

No. of class	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	LapLRR	NSHLRR	PGSL	AWNLRR
2	97.50	**99.88**	**99.88**	99.75	99.75	99.25	**99.88**	99.63	97.38	**99.88**
4	90.88	98.69	98.69	98.31	94.56	91.19	98.94	99.00	96.73	**99.19**
6	76.83	86.87	84.48	93.54	87.37	66.75	90.73	94.35	71.96	**96.00**
8	80.78	88.82	88.44	89.44	84.31	71.25	87.65	90.25	89.41	**92.33**
10	64.75	83.23	82.57	79.12	67.39	65.78	75.34	82.18	82.28	**84.26**

Note: bold numbers denote the best results.

Table 7
Clustering Acc (%) of different methods on remaining real datasets.

Dataset	*K*-means	Rcut	Ncut	SSC	LRR	LatLRR	LapLRR	NSHLRR	PGSL	AWNLRR
MNIST	55.30	68.80	69.97	53.65	54.55	42.01	63.65	60.24	67.50	**72.34**
AR	31.19	48.39	48.52	64.73	56.37	57.14	65.19	65.75	47.13	**70.94**
LFW	22.18	23.79	24.03	29.29	23.81	25.08	27.49	27.98	23.04	**31.18**
MSRA	50.70	57.42	57.42	60.98	63.79	60.57	62.89	63.41	59.76	**67.58**
Cars	54.59	63.01	63.11	62.00	62.50	67.96	64.09	62.34	60.69	**68.62**
Vehicle	45.86	46.53	46.64	44.92	45.75	46.57	45.89	45.98	43.12	**48.50**
Yeast	31.79	34.42	34.67	39.34	36.93	37.40	37.42	40.74	36.05	**43.44**

Note: bold numbers denote the best results.

the graph learning. In other words, exploiting the distance relationships of data to regularize the graph has potential to learn a more reasonable and discriminative graph so that a better clustering performance can be obtained.

(4) From the experimental results of different methods on the YaleB database (Table 4 and Fig. 6(b)), we can find that all of the representation based methods achieve much better performance than those of methods that only exploit the distance information, *i.e.*, Ncut and Rcut. This illustrates that the representation based metrics are more robust than distance based metric in capturing the intrinsic structure of data under the condition of various illuminations. This indicates that the represen-

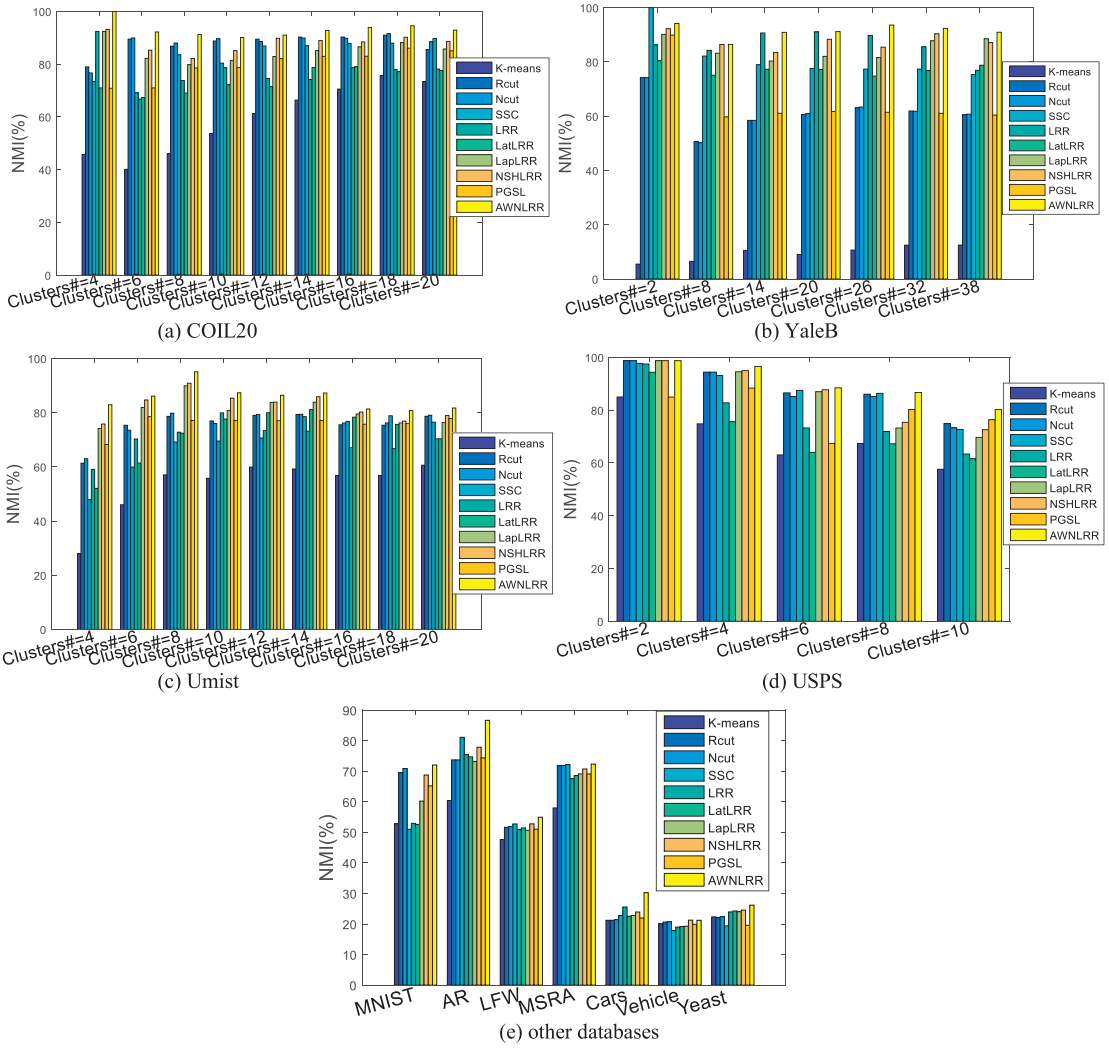


Fig. 6. Clustering NMIs (%) of different clustering methods on the (a) COIL20, (b) YaleB, (c) Umist, (d) USPS databases, and (e) other databases.

tation structures of data are also useful and contain discriminative information for data clustering.

(5) We should point out that both of the NSHLRR and the proposed method not only take into account the global and local structures of data, but also learns a sparsity graph. However, from the comparison of the two methods, it is obvious that the proposed method perform much better than NSHLRR. This is mainly because that the proposed method has potential to learn a more robust graph by introducing a weighted matrix to adaptively reinforce the role of important features and simultaneously reduce the role of those redundant features during graph learning.

From the above analyses, we can conclude that: (1) both distance relationships and representation relationships of data all contain discriminative information; (2) the optimal graph can be learned if and only if the two structures can be effectively exploited. Compared with other methods, the proposed method has potential to learn a more robust graph than other methods owing to its effective in uncovering the important features and improving their roles during graph learning. The above experimental results also prove the superiority of the proposed method which is analyzed in the Section 4.3.

5.3. Analysis of the graph initialization and parameter selection

From Algorithm 1, there are four uncertainty parameters, *i.e.*, balanced parameters λ_1, λ_2, λ_3, and initialized nearest neighbor number k. In this subsection, we will analyze the sensitivity of these parameters to the proposed method. Fig. 7 shows the clustering accuracy (%) *versus* the number of initial nearest neighbor size on the whole COIL20 and YaleB databases with the fixed parameters of λ_1, λ_2, λ_3. In Fig. 7(b), the maximum and minimum accuracies are 88.99% and 88.81% when nearest neighbor sizes are 6 and 8, respectively. The error between the maximum and minimum accuracies on the YaleB database is very small, *i.e.*, 0.18%. By the way, the maximum accuracy error on the COIL20 database is also very small, *i.e.*, 0.23%. Therefore, we can conclude that the clustering performance are very insensitive to the selection of nearest neighbor size of the initial graph Z. The major factor leads to this good property is that the proposed method can adaptively select nearest neighbors for each sample in the stage of graph learning. In the above experiments, we uniformly set the nearest neighbor size to 10.

Next we analyze the sensitivity of the three balanced parameters, *i.e.*, λ_1, λ_2, and λ_3 to the proposed method. Specially, λ_1 controls the values of weighted matrix S. This term is used to avoid the trivial solution to S. Usually, a small value such as {0.1, 0.01, 0.001} is a proper value to λ_1. Figs. 8 and 9 show the clustering

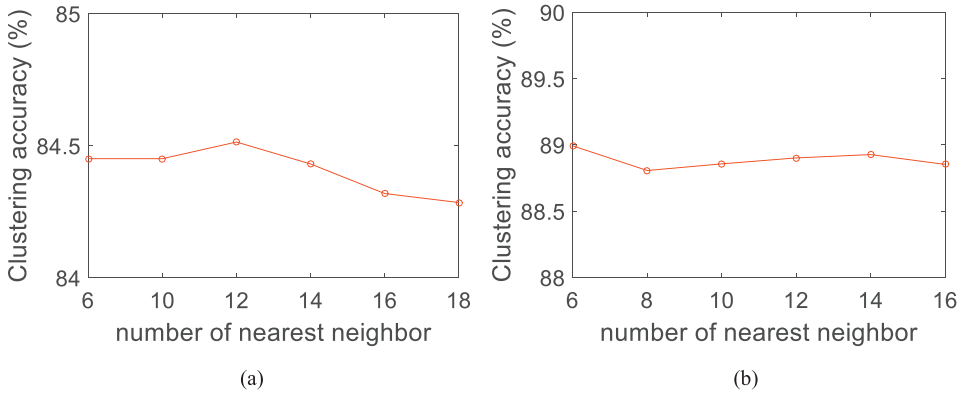


Fig. 7. Clustering accuracy (%) viruses the number of initial nearest neighbor size on the (a) COIL20 database and (b) YaleB database.

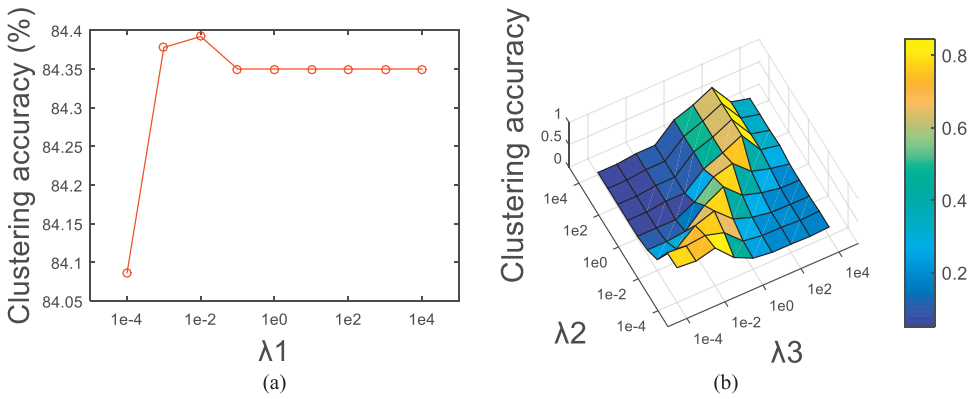


Fig. 8. Clustering accuracy *versus* different values of (a) parameters λ_1 when λ_2 and λ_3 are fixed, (b) parameters λ_2 and λ_3 when $\lambda_1 = 0.01$ on the COIL20 dataset. .

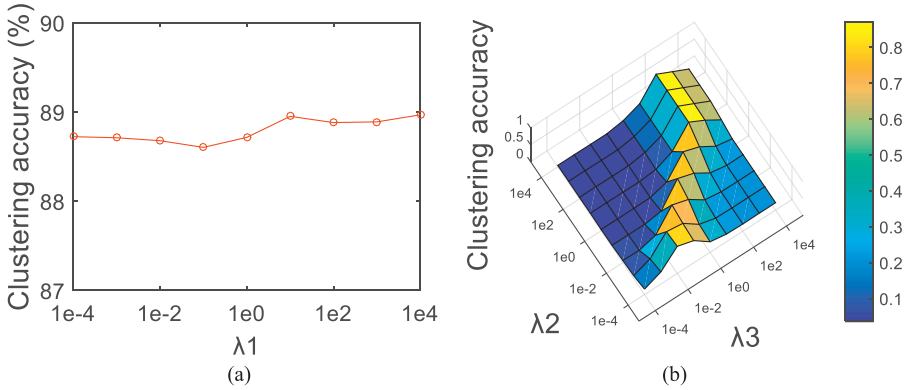


Fig. 9. Clustering accuracy *versus* different values of (a) parameters λ_1 when λ_2 and λ_3 are fixed, (b) parameters λ_2 and λ_3 when $\lambda_1 = 0.01$ on the YaleB dataset.

accuracies of the proposed method with respect to the three parameters on the COIL20 and YaleB databases. From Figs. 8 and 9, one can see that when parameters λ_2 and λ_3 are fixed, the clustering performance is insensitive to the selection of parameter λ_1 in the range of $[10^{-4}, 10^4]$. Compared with parameter λ_1, the clustering performance is very sensitive to parameters λ_2 and λ_3. This is mainly because they are regularized on the graph and directly determine the roles of corresponding terms during graph learning. For example, a small λ_2 will lead to a denser graph which may produce a bad performance. So it is necessary to tune suitable values for these three parameters to obtain a satisfactory performance.

Due to the diversity of databases, it is difficult to find the common values of these three parameters for different databases. Here we present a simple and effective way to find their optimal values. According to previous analysis, we can first simply fix parameter λ_1 to a small value such as 0.01, then find the candidate combination of parameters λ_2 and λ_3 from the coarse set of $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4\}$. According to the obtained best combination of these two parameters, we can further define a fine candidate set for these two parameters that the optimal values may be exist. Then we perform the method again with different combinations of these two parameters selected from the fine candidate range. In this way, we can finally obtain the optimal parameters for λ_2 and λ_3 so that the best clustering performance is guaranteed.

5.4. Experiments in dealing with new sample

Following the experimental settings in 1, we chose two large-scale datasets, *i.e.*, PenDigits [59] and Covtype [60], to evaluate the

Table 8
Performances of different methods on the PenDigits and Covtype datasets. Bold numbers denote the best results.

Dataset	PenDigits			Covtype			
Algorithm	Acc (%)	NMI (%)	Time (s)	Acc (%)	NMI (%)	Time (s)	Computational complexity
K-means	68.38	65.88	–	27.03	5.80	–	–
Rcut	71.73	67.86	0.34	29.07	5.50	0.12	$O(n^2)$
Ncut	74.39	69.25	0.40	30.67	5.11	0.15	$O(n^2)$
SSC	76.59	70.96	577.6	27.13	5.55	223.01	$O(\tau mn^3)$
LRR	76.21	67.91	7.13	30.14	6.71	1.48	$O(\tau(d^2n+d^3))$, $(d\le n)$
LatLRR	76.08	69.97	7.24	29.70	6.40	1.79	$O(\tau(d^2n+d^3))$, $(d\le n)$
LapLRR	77.29	71.44	59.66	29.14	5.32	34.83	$O(\tau n^3)$
NSHLRR	77.89	71.56	168.21	29.71	6.18	67.52	$O(\tau rn^2)$, $(r\le n)$
PGSL	75.91	71.05	43.43	32.26	6.18	17.49	$O(\tau(n^3+dn^2))$, $(d\le n)$
AWLRR	**80.11**	**73.87**	35.60	**34.30**	**7.68**	11.46	$O(\tau rn^2)$, $(r\le n)$

Note: These methods all use the same approaches, *i.e.*, spectral clustering and CRC, to obtain the final clustering and classification results, which have the same computational cost in different methods, thus we report only the running time of different methods in graph learning.

effectiveness of the proposed approach in dealing with new samples. PenDigits is a handwritten digit feature dataset which contains 10,992 samples and 10 classes. Each sample in the PenDigits dataset has 16 features. The Covtype dataset is created for predicting forest cover types from cartographic variables. It is composed of 581,012 samples provided by 7 classes, in which each sample is represented by 54 features. For the two datasets, we randomly select 100 samples from each class as training set (in-sample) and treat the remaining samples as test set (out-sample), respectively. We first perform different clustering methods on the training set to obtain their corresponding clustering results and then use Algorithm 3 to recognize the out-sample. All experiments are conducted 10 times in the same hardware and software platforms and the mean clustering accuracies (Acc) (%) and NMI (%) are reported for comparing.

Table 8 shows the experimental results of different methods in recognizing the new samples and clustering the in-samples. Besides, the running times of different methods are also reported. It is obvious that the proposed method outperforms the other methods in terms of the Acc and NMI on these two datasets. This also proves the superiority of the proposed method in dealing with the new sample. Moreover, from the comparison of the computational complexity and running time, we can find that the running time is generally consistent with the computational complexity. The running time and the computational complexity of SSC are much higher than the other methods. Although the computational complexity of NSHLRR is lower than LapLRR and PGSL, its running time is higher than that of the two methods. This is mainly because NSHLRR needs more iteration steps than the other two methods to find the optimal solution. Compared with the similar graph learning methods, *i.e.*, LapLRR, SSC, and NSHLRR, the proposed method is more efficient.

6. Conclusions

In this paper, a novel graph learning method called adaptive weighted nonnegative low-rank representation is proposed to learn the intrinsic graph for data clustering. By introducing an adaptive weighted matrix to constrain the self-representation term, the role of those redundant features especially the noise and outliers can be effectively reduced so that a more robust graph can be obtained. Compared with other methods, the proposed method simultaneously captures the global representation structure and local geometric structure of data by integrating the distance regularization term into the LRR model, and thus can learn a more discriminative graph for data clustering. Experimental results on both synthetic and real databases including face, handwritten digital, object, and non-image databases show that the proposed method achieves the best performance than other state-of-the art methods.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant no. 61332011, Guangdong Province high-level personnel of special support program under Grant no. 2016TX03X164, National Natural Science Foundation of Guangdong Province under Grant no. 2017A030313384, Shenzhen Fundamental Research fund under Grant no. JCYJ20160331185006518, and Economic, Trade & Information Commission of Shenzhen Municipality under Grant no. 20170504160426188.

Reference

[1] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[2] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[3] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient k-means clustering algorithm: analysis and implementation, IEEE Trans. Pattern Anal. Mach. Intel. 24 (7) (2002) 881–892.
[4] F. Murtagh, A survey of recent advances in hierarchical clustering algorithms, Comput. J. 26 (4) (1983) 354–359.
[5] N.R. Pal, J.C. Bezdek, C.K. Tsao, Generalized clustering networks and Kohonen's self-organizing scheme, IEEE Trans. Neural Netw. 4 (4) (1993) 549–557.
[6] U.V. Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[7] X. Peng, J. Feng, J. Lu, W.-Y. Yau, Z. Yi, Cascade subspace clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 2478–2484.
[8] F. Nie, H. Huang, Subspace clustering via new low-rank model with discrete group structure constraint, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 1874–1880.
[9] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intel. 22 (8) (2000) 888–905.
[10] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[11] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, A survey of sparse representation: algorithms and applications. IEEE Access, 3: 490–530.
[12] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, N. Yu, Non-negative low rank and sparse graph for semi-supervised learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2328–2335.
[13] R. Vidal, P. Favaro, Low rank subspace clustering (LRSC), Pattern Recognit. Lett. 43 (2014) 47–61.
[14] X. Fang, Y. Xu, X. Li, Z. Lai, W.K. Wong, Robust semi-supervised subspace clustering via non-negative low-rank representation, IEEE Trans. Cybernet. 46 (8) (2015) 1828–1838.
[15] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intel. 35 (11) (2013) 2765–2781.
[16] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intel. 35 (1) (2013) 171–184.
[17] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, S. Zhan, Low-rank preserving projection via graph regularized reconstruction, IEEE Trans. Cybernet. (2018), doi:10.1109/TCYB.2018.2799862.
[18] M. Yin, J. Gao, Z. Lin, Laplacian regularized low-rank representation and its applications, IEEE Trans. Pattern Anal. Mach. Intel. 38 (3) (2016) 504–517.

[19] J. Liu, Y. Chen, J. Zhang, Z. Xu, Enhancing low-rank subspace clustering by manifold regularization, IEEE Trans. Image Process. 23 (9) (2014) 4022–4030.
[20] K. Tang, R. Liu, Z. Su, Z. Jie, Structure-constrained low-rank representation, IEEE Trans. Neural Netw. Learn. Syst. 25 (12) (2014) 2167–2179.
[21] J. Feng, Z. Lin, H. Xu, S. Yan, Robust subspace segmentation with block-diagonal prior, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3818–3825.
[22] Y. Jian, L. Lei, J. Qian, T. Ying, F. Zhang, X. Yong, Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes, IEEE Trans. Pattern Anal. Mach. Intel. 39 (1) (2014) 156–171.
[23] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, Z. Yi, Deep subspace clustering with sparsity prior, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 1925–1931.
[24] L. Fei, Y. Xu, X. Fang, J. Yang, Low rank representation with adaptive distance penalty for semi-supervised subspace classification, Pattern Recognit. 67 (2017) 252–262.
[25] Y. Xu, Z. Zhong, J. Yang, J. You, D. Zhang, A new discriminative sparse representation method for robust face recognition via l_2 regularization, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2017) 2233–2242.
[26] Y. Xu, X. Li, J. Yang, Z. Lai, D. Zhang, Integrating conventional and inverse representation for face recognition, IEEE Trans. Cybernet. 44 (10) (2014) 1738–1746.
[27] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intel. 31 (2) (2009) 210–227.
[28] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition? in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 471–478.
[29] Y. Meng, Z. Lei, Y. Jian, Z. David, Regularized robust coding for face recognition, IEEE Trans. Image Process. 22 (5) (2013) 1753–1766.
[30] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.
[31] Y. Xu, Z. Zhang, G. Lu, J. Yang, Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification, Pattern Recognit. 54 (C) (2016) 68–82.
[32] J. Zheng, P. Yang, S. Chen, G. Shen, W. Wang, Iterative *re*-constrained group sparse face recognition with adaptive weights learning, IEEE Trans. Image Process. 26 (5) (2017) 2408–2423.
[33] J. Liu, S. Ji, J. Ye, SLEP: sparse learning with efficient projections, Ariz. State Univ. 6 (491) (2009) 1–61.
[34] F. Nie, X. Wang, M.I. Jordan, H. Huang, The constrained Laplacian rank algorithm for graph-based clustering, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1969–1976.
[35] R. He, W.S. Zheng, B.G. Hu, X.W. Kong, Nonnegative sparse coding for discriminative semi-supervised learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 2849–2856.
[36] L. Fei, Y. Xu, B. Zhang, X. Fang, J. Wen, Low-rank representation integrated with principal line distance for contactless palmprint recognition, Neurocomputing 218 (2016) 264–275.
[37] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.
[38] L. Hagen, A.B. Kahng, New spectral methods for ratio cut partitioning and clustering, IEEE Trans. Comput-Aided Des. Integr. Circuits Syst. 11 (9) (2006) 1074–1085.
[39] X. Peng, L. Zhang, Z. Yi, Inductive sparse subspace clustering, Electron. Lett. 49 (19) (2013) 1222–1224.
[40] M. Yin, J. Gao, Z. Lin, Q. Shi, Y. Guo, Dual graph regularized latent low-rank representation for subspace clustering, IEEE Trans. Image Process. 24 (12) (2015) 4918–4933.
[41] X. Peng, H. Tang, L. Zhang, Z. Yi, S. Xiao, A unified framework for representation-based subspace clustering of out-of-sample and large-scale data, IEEE Trans. Neural Netw. Learn. Syst. 27 (12) (2016) 2499–2512.
[42] L. Zhang, M. Yang, X. Feng, Y. Ma, and D. Zhang, Collaborative Representation based Classification for Face Recognition. Technical report. arXiv:1204.2358, 2012.
[43] E. Esser, Applications of Lagrangian-based alternating direction methods and connections to split Bregman, CAM Rep. 9 (2009) 31.
[44] Z. Lin, R. Liu, H. Li, Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning, Mach. Learn. 99 (2) (2015) 287–325.
[45] Z. Zhang, Y. Xu, L. Shao, J. Yang, Discriminative block-diagonal representation learning for image recognition, IEEE Trans. Neural Netw. Learn. Syst. 1 (2017) 1–16.
[46] Z. Lin, M. Chen, and Y. Ma, The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. UIUC Technical Report UILU-ENG-09-2215, UIUC, 2009: arXiv:1009.5055.
[47] R. Glowinski, P. Le Tallec, Augmented Lagrangian and operator-splitting methods in nonlinear mechanics, SIAM (1989).
[48] J. Eckstein, D.P. Bertsekas, On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators, Math. Program. 55 (1) (1992) 293–318.
[49] G. Liu, S. Yan, Latent low-rank representation for subspace segmentation and feature extraction, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 1615–1622.
[50] Q. Mao, L. Wang, I. Tsang, Y. Sun, Principal graph and structure learning based on reversed graph embedding, IEEE Trans. Pattern Anal. Mach. Intel. 39 (11) (2017) 2227–2241.
[51] J. Huang, F. Nie, H. Huang, A new simplex sparse learning model to measure data similarity for clustering, in: Proceedings of the International Conference on Artificial Intelligence, 2015, pp. 3569–3575.
[52] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: Proceedings of the International Conference on Machine Learning, 2010, pp. 663–670.
[53] M. Lichman, UCI Machine Learning Repository, School of Information and Computer Sciences, University of California, 2013 .http://archive.ics.uci.edu/ml.
[54] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intel. 23 (6) (2001) 643–660.
[55] A.M. Martinez, The AR Face Database, CVC, 1998 CVC Technical Report 24.
[56] Daniel B Graham, and N.M. Allinson, Face recognition: from theory to applications. NATO ASI Ser. F Comput. Syst. Sci., 1998, 163: 446–456.
[57] S.K. N. a. H. M. S.A. Nene, Columbia Object Image Library (COIL-20). Technical Report CUCS-005-96, CUCS, 1996: 1–6.
[58] E. Learned-Miller, G.B. Huang, A. RoyChowdhury, H. Li, G. Hua, Labeled faces in the wild: A survey, Advances in face detection and facial image analysis, Springer, Cham. (2016) 189–248.
[59] F. Alimoglu, E. Alpaydin, Combining multiple representations and classifiers for pen-based handwritten digit recognition, in: Proceedings of the International Conference on Document Analysis and Recognition, 1997, pp. 637–640.
[60] J.A. Blackard, D.J. Dean, Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables, Comput. Electron. Agric. 24 (3) (1999) 131–151.

Jie Wen received the M.S. degree at Harbin Engineering University, China in 2015. He is currently pursuing the Ph.D. degree in computer science and technology at Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. His research interests include, image and video enhancement, pattern recognition and machine learning.

Bob Zhang received the B.A. degree in computer science from York University, Toronto, ON, Canada, in 2006, the M.A.Sc. degree in information systems security from Concordia University, Montreal, QC, Canada, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2011. After graduating from Waterloo, he remained with the Center for Pattern Recognition and Machine Intelligence, and later was a Postdoctoral Researcher in the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently an Assistant Professor in the Department of Computer and Information Science, University of Macau, Taipa, Macau. His research interests focus on biometrics, pattern recognition, and image processing. Dr. Zhang is a Technical Committee Member of the IEEE Systems, Man, and Cybernetics Society, an Associate Editor for the International Journal of Image and Graphics, as well as an Editorial Board member for the International Journal of INFORMATION.

Yong Xu was born in Sichuan, China, in 1972. He received the Ph.D. degree in Pattern recognition and Intelligence System at the Nanjing University of Science and Technology (NUST) in 2005. Now, he works at Shenzhen Graduate School, Harbin Institute of Technology. His current interests include pattern recognition, biometrics, machine learning and video analysis. More information please refer to http://www.yongxu.org/lunwen.html.

Jian Yang received the B.S. degree in mathematics from Xuzhou Normal University, Xuzhou, China, in 1995, the M.S. degree in applied mathematics from Changsha Railway University, Changsha, China, in 1998, and the Ph.D. degree in the subject of pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002. In 2003, he was a Post-Doctoral Researcher with the University of Zaragoza, Zaragoza, Aragon, Spain. From 2004 to 2006, he was a Post-Doctoral Fellow with the Biometrics Center, Hong Kong Polytechnic University, Kowloon, Hong Kong. From 2006 to 2007, he was a Post-Doctoral Fellow with the Department of Computer Science, New Jersey Institute of Technology, Newark. Currently, he is a Professor with the School of Computer Science and Technology, NUST. He is the author of more than 50 scientific papers on pattern recognition and computer vision. His journal papers have been cited more than 1200 times on the ISI Web of Science, and 2000 times on Google Scholar. His current research interests include pattern recognition, computer vision, and machine learning.

Na Han received her B.S. degree in computer science and technology at HIT in 2004. She is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Technology, Guangdong University of Technology. His current research interests include pattern recognition and machine learning.