Pattern Recognition 112 (2021) 107736

Contents lists available at ScienceDirect

Pattern Recognition

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

A hierarchical weighted low-rank representation for image clustering and classification

Zhiqiang Fu[a,b], Yao Zhao[a,b,*], Dongxia Chang[a,b], Yiming Wang[a,b]

[a] Institute of Information Science, Beijing Jiaotong University, Beijing, 100044, China
[b] Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, 100044, China

ARTICLE INFO

Article history:
Received 23 May 2020
Revised 25 August 2020
Accepted 29 October 2020
Available online 4 November 2020

Keywords:
Low-rank representation
Clustering
Semi-supervised learning
Similarity graph construction

ABSTRACT

Low-rank representation (LRR), which is a powerful method to find the low-dimensional subspace structure embedded in high-dimensional data spaces, has been used in both unsupervised learning and semi-supervised classification. LRR aims at finding the lowest rank representation that can express each data sample as linear combination of other samples. However, this method doesn't consider the geometrical structure of the data. Thus the similarity and local structure might be lost in the process of learning. Motivated by this, a novel hierarchical weighted low-rank representation (HWLRR) is proposed in this paper. In the new algorithm, a hierarchical weighted matrix is defined to find more samples that may belong to the same subspace using affinity propagation. By taking advantage of the affinity propagation, our proposed method can preserve both local and global structure of the whole dataset. The experimental results on both unsupervised learning and semi-supervised classification demonstrate the superiority of our proposed method.

1. Introduction

Low-rank representation (LRR) [1,2] is a promising learning method and aims to find a representation with the lowest rank for a given dataset. Because of its excellent performance in capturing the low-dimensional structure embedded in the data space, it has been extensively used in image classification [3], image clustering [4], face recognition [5], feature extraction [6], subspace segmentation [7], face sketch synthesis[8] and metric learning [9].

As mentioned in matrix completion and low-rank matrix recovery, it is always assumed that the data points usually linearly correlate with each other in the subspace if they are drawn from the same pattern. Therefore, data points sampled from different clusters can be regarded as samples nearly drawn from a union of multiple low-rank subspaces. Consequently, it is desirable to develop a technique to find the intrinsic geometrical subspace structure of a given data. In fact, numerous low-rank learning methods have been developed in the past. Among them, LRR and robust principal component analysis (RPCA) [10] are two typical methods. RPCA assumes that the embedded data structure lies in just one low-rank subspace. However, it neglects the individual subspace's specific property [11]. By contrast, LRR method explicitly considers the data structure approximately embeded in several low-dimensional subspaces.

LRR is an effective way to learn the low-dimensional structure based on the assumption that the data approximately spans several low-dimensional subspaces, so plenty of methods based on LRR have been proposed to solve clustering and classification problems. In Ref[1]., LRR was developed to cluster the samples into their respective subspaces and remove possible outliers. In this method, the authors chose the observed data itself as the dictionary. However, the performance could depress if the observations were insufficient or corrupted [2]. To overcome this shortcoming, Latent LRR [2] was proposed to construct the dictionary using both unobserved and observed data. In fact, noiseless Latent LRR couldn't obtain a unique solution. To address this issue, the robust Latent LRR was put forward, which choses the sparsest matrix from the solution set of Latent LRR [12]. To improve the computational speed, Liu et al.[13] introduced a fixed-rank strategy to reveal the structure of multiple subspaces in closed-form. Moreover, Fang et al. [14,15] combined LRR with semi-supervised clustering as a solution to classification.

Conventional LRR algorithms usually capture the subspace by learning the global structure of subspace but miss the local structure. However, to find the geometric structure embedded in the data, LRR should capture not only global relevance but also local geometric structure. To preserve the geometrical structure, many researchers have considered learning from manifold learning meth-

* Corresponding author.
E-mail address: yzhao@bjtu.edu.cn (Y. Zhao).

https://doi.org/10.1016/j.patcog.2020.107736
0031-3203/

ods, e.g., ISOMAP [16], Laplacian Eigenmap [17] and Neighborhood Preserving Embedding (NPE) [18].

Motivated by these manifold learning methods, some modified LRR methods have been proposed. Zheng et al. [19] proposed a graph regularized sparse coding aiming to take into account the local manifold of the data and learn sparse representations. To remove striping noise in hyperspectral images, Lu et al. [20] proposed a novel graph-regularized LRR (GLRR) and improved their approach by adopting graph technique. Similarly, a general Laplacian regularized low-rank representation framework for data representation was given in Ref[21]. By making use of graph regularizer, this method could not only capture global low-dimensional structure but also represent the intrinsic geometric information in data. Moreover, some recent evidences show that graph is a good choice for capturing the structure embedded in high-dimensional space [21].

Most modified LRR methods mentioned above use k-nearest neighbor (kNN) algorithm to improve their performance. However, they ignore the structure of the higher hierarchy neighbors, i.e., neighbors' neighbors. Moreover, the number of neighbors k affects the performance of the algorithm and kNN may bring bad effect for representation [22]. Motivated by this, we propose a hierarchical weighted low-rank representation (HWLRR) method. In this new algorithm, a hierarchical weighted matrix is defined to balance the contribution of the k-nearest neighbors and the higher hierarchy neighbors. Therefore, the representation produced by HWLRR can better capture the local intrinsic structure and learn the global structure of the dataset. Moreover, this representation can work well with both clustering and classification algorithms (e.g., SC [23] and GFHF [24]). Experiments show that the proposed algorithm has good results in image clustering and classification.

The remainder of our paper is organized as follows. In Section 2, we briefly review some related works of LRR. Section 3 provides some definitions necessary for our approach. Then, the proposed HWLRR is introduced in Section 4. Furthermore, the experimental results for image clustering and classification are given in Section 5. Finally, Section 6 concludes our paper.

2. Related works

LRR [1] has attracted much attention due to its overwhelming advantages in a wide range of real-world applications. It is better at learning the global structure, because it obtains a representation under the global low-rank constraint [25]. LRR is based on the assumption that each data sample in a given dataset can be regarded as sampling from some independent low-dimensional subspaces. Therefore, it is an effective way to find data points that belong to the same subspace.

For a given dataset Y, Y is sampled from a combination of low dimensional subspace $U_{j=1}^{k}S_j$, where S_1, S_2, S_3,...,S_k are low-dimensional subspaces. We assume that each subspace is independent and has no noise, then the low-dimensional subspace can be represented by the block distribution of the matrix. And the low-rank representation can be expressed as follows,

$$\min_{Z} \ rank(Z), \ s.t. Y = AZ \tag{1}$$

where $rank(\cdot)$ is the rank of a matrix, A is the dictionary used to represent Y, Z is the representation matrix in which each column z_i is the coefficient vector of y_i based on the dictionary matrix A. The low-rank constraint can ensure a strong correlation among the samples in the same subspace. But in practical application, noise is unavoidable. To improve the robustness, the low-rank representation model is modified as

$$\min_{Z,E} \ rank(Z) + \lambda \|E\|_l, \ s.t. Y = AZ + E \tag{2}$$

where λ is a penalty parameter to balance the low-rank term and the reconstruction error, E is the error fitting matrix, and $\|\cdot\|_l$ is the regularization of noise. The LRR theory shows that minimizing the rank of the data can be solved by minimizing the nuclear norm, and the low-rank representation problem can be expressed as the following minimization problem,

$$\min_{Z,E} \ \|Z\|_* + \lambda \|E\|_l, \ s.t. Y = AZ + E \tag{3}$$

where $\|\cdot\|_*$ is the nuclear norm defined as the sum of all singular values of the matrix. By using an appropriate dictionary A, the intrinsic structure of the data can be captured. Normally the data matrix Y is used as the dictionary [26], then the general LRR model can be expressed as

$$\min_{Z,E} \ \|Z\|_* + \lambda \|E\|_l, \ s.t. Y = YZ + E \tag{4}$$

In order to improve the performance of LRR, plenty of efforts have been adopted by imposing a penalty on Z and E. Thus different kinds of LRR-based methods were developed. To better handle the modified LRR method, a general rank minimization problem can be formulated as

$$\min_{Z,E} \ \|Z\|_* + Q(Z, E), \ s.t. Y = YZ + E \tag{5}$$

where Q is a penalty function about Z and E. In fact, the penalty function $Q(Z, E)$ shown in Eq. (5) is the most important part of the LRR model. A number of methods have been proposed to construct it. For example, Yin et al.[21] defined $Q(E, Z) = \lambda \|Z\|_1 + \beta tr\left(ZLZ^T\right) + \gamma \|E\|_1$ to learn the manifold structure within the data.

3. Hierarchical weighted matrix

In this section, some definitions needed in the next section are given. For a given dataset $Y = [y_1, y_2, \cdots, y_n]$, it is usually assumed that the relationship between a data point and its neighbors is linear [27]. Thus, the local geometry of all the data points can be effectively characterized by the linear combination of its neighbors, and k-nearest neighbors [28] are usually used. Intuitively, the data point and its k-nearest neighbors belong to the same subspace if k is small. This assumption has been widely accepted and has been used in many algorithms [22,29]. But it is difficult to give an appropriate k. If k is too small, some samples in the same subspace can be lost. In fact, not only the nearest neighbors but also their neighbors probably belong to the same subspace. In order to make full use of the neighbors, a hierarchical weighted matrix is proposed in this section. To further illustrate it, some key definitions are given as follows.

Definition 1. (First Order Adjacency Matrix). Given a dataset Y, the first order adjacency matrix W can be constructed by the k-nearest neighbors based on Euclidean distance. Here $w_{i,j}$ shows whether there is a neighborhood relationship between y_i and y_j and it is defined as follows,

$$w_{i,j} = \begin{cases} 1, & y_j \in N_k(y_i) \\ 0, & otherwise \end{cases} \tag{6}$$

where $N_k(y_i)$ is the k-nearest neighbors set of data y_i.

Definition 2. (First Hierarchy Neighbor). Given a dataset $Y = [y_1, y_2, \cdots, y_n]$, W is the first order adjacency matrix obtained by Definition 1. If $w_{i,j} = 1$, then y_j is defined as the first hierarchy neighbor of y_i.

Here, for the given dataset showed in Fig. 1, each point can find its two nearest neighbors with $k = 2$ as shown in Fig. 2. The first hierarchy neighbors of Point 1 and Point 8 are given in Fig. 3. It is clear that many data points that belong to the same subspace are lost by the first hierarchy neighbors. To address this issue, the

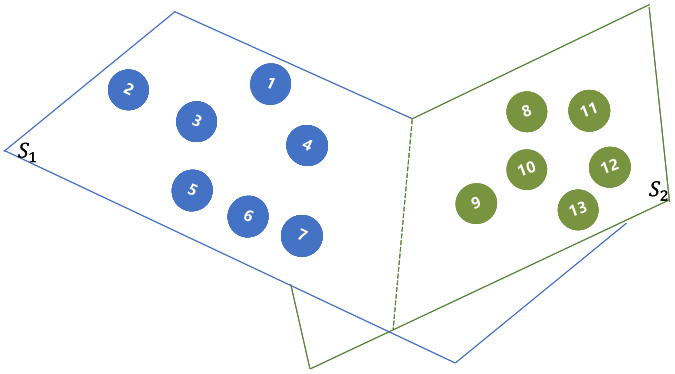

Fig. 1. Dataset Y. Here the points with the same color belong to the same subspace. S_1 and S_2 denote two different subspaces.

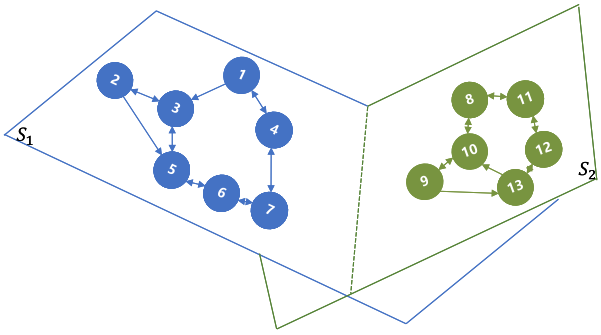

Fig. 2. The k-nearest neighbors of dataset Y with $k = 2$.

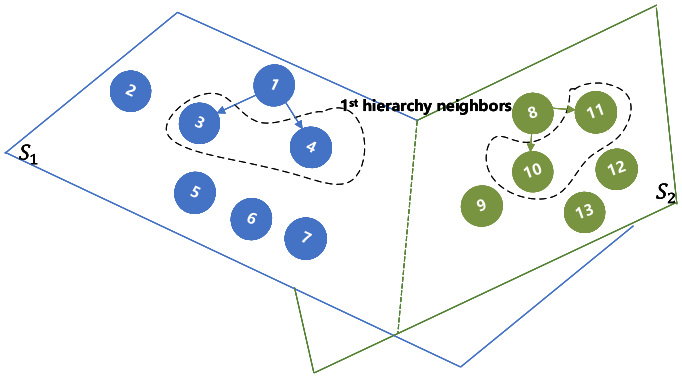

Fig. 3. The first hierarchy neighbors of Point 1 and Point 8 ($k = 2$). Here Point 3 and Point 4 are the 2-nearest neighbors of Point 1. Point 10 and Point 11 are the first hierarchy neighbors of Point 8. However, many points in the same subspace are missed by the first hierarchy neighbors.

higher hierarchy neighbor is defined based on the first hierarchy neighbor as following.

Definition 3. (Higher Hierarchy Neighbor). For a data point y_i, y_j is defined as y_i's $(l+1)$-th hierarchy neighbor if y_j is its l-th hierarchy neighbor's neighbor.

According to this definition, the second hierarchy neighbors of Point 1 and Point 8 can be found easily and are shown in Fig. 4. It is obvious that the higher hierarchy neighbors are more than the first hierarchy neighbors, and more neighbors can preserve more data structure. As Fig. 5 shows, we can find all the neighbors on different hierarchies for Point 1 and Point 8.

Definition 2 and Definition 3 give the definitions of the higher hierarchy neighbors based on kNN. For a given data point, we define that its neighbors should have equal importance if they lie

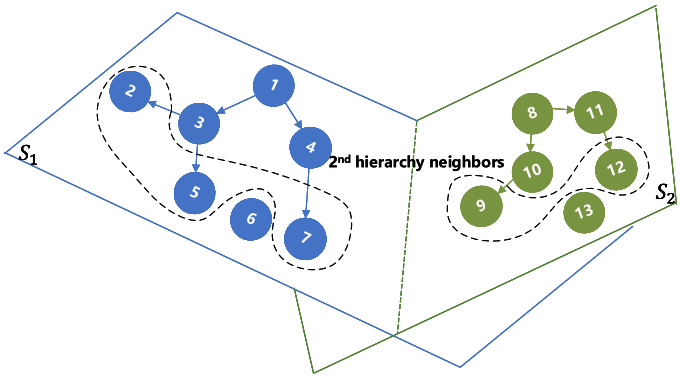

Fig. 4. The second hierarchy neighbors of Point 1 and Point 8 ($k = 2$).

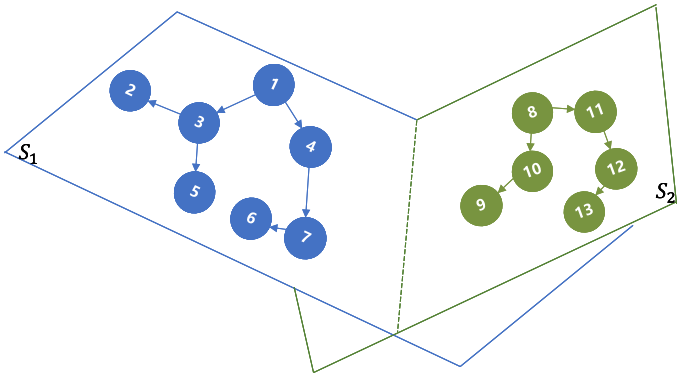

Fig. 5. All the hierarchy neighbors of Point 1 and Point 8.

on the same hierarchy, but have different importance if they lie on different hierarchies. Motivated by Markov process [30], we assume that each point can transfer to another, and the transferring probability is defined.

Definition 4. (First Order Transfer Probability). For a data point y_i, the first order transfer probability to its first hierarchy neighbors is defined as

$$p = \frac{1}{k} \tag{7}$$

The first order transfer probability given in Definition 4 measures the probability from a point to its first hierarchy neighbors. As there exist higher hierarchy neighbors, the higher order transfer probability is defined in the following.

Definition 5. (Higher Order Transfer Probability). For a data point y_i, y_j is the higher hierarchy neighbor of y_i. Then the higher order transfer probability from y_i to y_j is defined as follows,

$$p_{i,j} = \begin{cases} \left(\frac{1}{k}\right)^{l_j}, y_j \in H_i \\ 0, otherwise \end{cases} \tag{8}$$

where l_j is the ordinal of the hierarchy which y_j lies on and H_i is the set of all the high hierarchy neighbors of y_i.

Using the definition of the first order and higher order transfer probability, a completely reachable matrix can be defined by Definition 6.

Definition 6. (Completely Reachable Matrix). The completely reachable matrix P_s is a matrix composed of $p_{i,j}$ and represents the transfer probability of all the data samples. It is obtained from a recursive formula as

$$P_{i+1} = d\left(\frac{1}{k^{i+1}} \cdot f(f(P_iP_1) - f(P_i))\right) + P_i \tag{9}$$

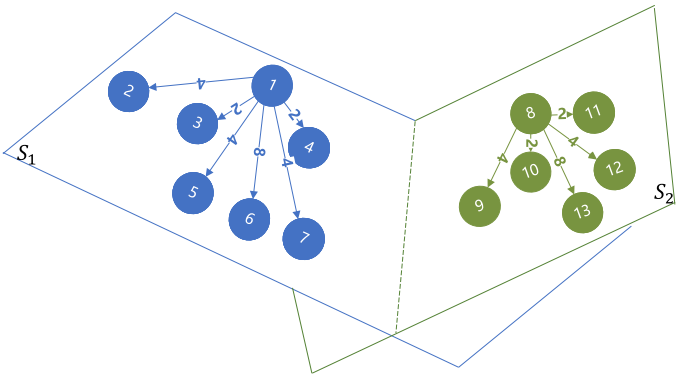

Fig. 6. The hierarchical weighted matrix T ($k = 2$). The value on the arrow is the weight of each point.

where $P_1 = W/k$, $d(\cdot)$ is a function defined as

$$d(X) = \begin{cases} 0, i = j \\ x_{i,j}, i \neq j \end{cases}$$

where X is a matrix and $x_{i,j}$ is the element on the ith row and jth column. $f(\cdot)$ is an approximate symbolic function which is defined as

$$f(X) = \begin{cases} 1, x_{i,j} > 0 \\ 0, x_{i,j} \leq 0 \end{cases}$$

the iteration stops when $P_{i+1} = P_i$ and a steady completely reachable matrix P_s is obtained. For convenience, a weighted matrix, hierarchical weighted matrix T, is defined in Definition 7 basing on the completely reachable matrix P_s.

Definition 7. (Hierarchical Weighted Matrix). The hierarchical weighted matrix T represents the cost of transfering from y_i to y_j and it is defined as

$$t_{i,j} = \begin{cases} \frac{1}{p_s(i,j)}, p_s(i,j) \neq 0 \\ \infty, p_s(i,j) = 0 \end{cases} \tag{10}$$

and when $p_s(i,j) = 0$, we use 10^6 to represent infinite. In the hierarchical weighted matrix T, the weight of a sample depends on the hierarchy that the sample lies on, as shown in Fig. 6, and the value on the arrow is the weight of each point.

The hierarchical weighted matrix T not only takes into account the nearby and similar data points, but also takes into account the faraway and less similar data points by affinity propagation.

4. Hierarchical weighted low-rank representation

Many LRR methods construct the penalty function of representation based on kNN, thus these methods generally use a fixed parameter k and may fail to capture global structure. However, as mentioned in [31], a good penalty function should reveal the intrinsic dimensionality [32] or complexity of the data by capturing the global structure embedded in the data. To improve the performance of the methods based on kNN, HWLRR is proposed in this section.

4.1. HWLRR

In general, LRR methods usually consist of the following two steps. First, the samples which may belong to the same subspace are found for each data point. Then, each data point is linearly represented by these samples with global low rank constraint.

In our HWLRR, the hierarchical weighted matrix proposed in Section 3 is used to find the samples which may belong to the same subspace. The hierarchical weighted matrix T not only learns the local structure based on the k-nearest neighbors, but also takes into account the global structure by affinity propagation. Thus, the following regularization is defined to determine the coefficients between corresponding pairs of samples,

$$\sum_{i,j} t_{i,j} z_{i,j} \tag{11}$$

where $t_{i,j}$ is the cost of transferring from y_i to y_j and Z is the LRR matrix as defined in Eq. (5). $z_{i,j}$ is the related coefficients between y_i and y_j. Minimizing Eq. (11) can assign a large coefficient to the edge between points with low cost, and attach a small coefficient to the edge between points with high cost. Combining Eq. (11) with LRR, the proposed HWLRR method is defined as follows,

$$\min_Z \ \|Z\|_* + \beta \sum_{i,j} t_{i,j} z_{i,j}, s.t.\ Y = YZ, Z \geq 0 \tag{12}$$

where Z is the low-rank representation matrix of the dataset. $\beta > 0$ is a balance parameter. Using matrix operations to replace the element operations, HWLRR can be reformulated as

$$\min_Z \ \|Z\|_* + \beta tr(\Theta(T \otimes Z)), s.t.\ Y = YZ, Z \geq 0 \tag{13}$$

where $tr(\cdot)$ is the trace of a matrix, Θ is a matrix whose elements are all 1 and $\otimes$ is Hadamard operator.

But in real application scenarios, noise always exists in data, and it will affect the data processing. Therefore, we should consider the presence of noise in data Y [33]. Here, we introduce the noise matrix E to fit the noise, and the model can be changed into

$$\min_{Z,E} \ \|Z\|_* + \lambda \|E\|_1 + \beta tr(\Theta(T \otimes Z)), s.t.\ Y = YZ + E, Z \geq 0 \tag{14}$$

where $\|\cdot\|_1$ is the l_1-norm and $\lambda > 0$ is a balance parameter.

To solve the HWLRR optimization problems, we first introduce a variable matrix J, then HWLRR can be rewritten as

$$\min_{Z,E,J} \ \|Z\|_* + \lambda \|E\|_1 + \beta tr(\Theta(T \otimes J)), s.t.\ Y = YZ + E, Z = J, J \geq 0 \tag{15}$$

The enhanced Lagrange function [34] can be used to solve this problem. The Lagrange function of the problem is expressed as follows,

$$\begin{aligned} &L(Z, E, J, M_1, M_2, \mu) \\ &= \|Z\|_* + \lambda \|E\|_1 + \beta tr(\Theta(T \otimes J)) \\ &+ \langle M_1, Y - YZ - E \rangle + \langle M_2, Z - J \rangle \\ &+ \frac{\mu}{2} \left(\|Y - YZ - E\|_F^2 + \|Z - J\|_F^2 \right) \end{aligned} \tag{16}$$

where M_1, M_2 are two Lagrange multiplier, $\mu > 0$ is penalty coefficient, $\langle \cdot, \cdot \rangle$ means inner product and $\| \cdot \|_F$ is the Frobenius norm of matrix. Simplifying Eq. (16), we get

$$\begin{aligned} &L(Z, E, J, M_1, M_2, \mu) \\ &= \|Z\|_* + \lambda \|E\|_1 + \beta tr(\Theta(T \otimes J)) + \\ &\frac{\mu}{2} (\|Y - YZ - E + \frac{M_1}{\mu}\|_F^2 + \|Z - J + \frac{M_2}{\mu}\|_F^2) \end{aligned} \tag{17}$$

The problem Eq. (17) is unconstrained, so we can fix other variables, and update Z, E, J as follows,

$$Z_{k+1} = \arg \min_Z \|Z\|_* + \frac{\mu}{2} \left(\left\| Y - YZ - E_k + \frac{M_{1\,k}}{\mu} \right\|_F^2 + \left\| Z - J_k + \frac{M_{2\,k}}{\mu} \right\|_F^2 \right) \tag{18}$$

$$E_{k+1} = \arg \min_E \ \|E\|_1 + \frac{\mu}{2} \left\| Y - YZ_k - E + \frac{M_{1\,k}}{\mu} \right\|_F^2 \tag{19}$$

$$J_{k+1} = \arg \min_{J \geq 0} \ \beta tr(\Theta(T \otimes J)) + \frac{\mu}{2} \left\| Z_k - J + \frac{M_{2\,k}}{\mu} \right\|_F^2 \tag{20}$$

For Eq. (18), suppose that $q(Z) = \frac{\mu}{2}(\left\|Y - YZ - E_k + \frac{M_{1\,k}}{\mu}\right\|_F^2 + \|Z - J_k + \frac{M_{2\,k}}{\mu}\|_F^2)$, then the following approximate formula can be obtained,

$$\begin{aligned} Z_{k+1} &= \arg\min_Z \|Z\|_* + q(Z_k) + \langle \nabla_Z q, Z - Z_k\rangle + \frac{\mu\eta_Z}{2}\left(\|Z - Z_k\|_F^2\right) \\ &= \arg\min_Z \|Z\|_* + \frac{\beta\eta_Z}{2}\|Z - Z_k + \frac{1}{\eta_Z}\left(-Y^T(Y - YZ - E_k + M_{1\,k}/\mu)\right) \\ &\quad + (Z - J_k + M_{2\,k}/\mu)\|_F^2 \end{aligned} \tag{21}$$

where $\nabla_Z q$ is the partial derivative of q relative to Z. $\eta_Z = 2\|Y\|_2$ is a constant, $\|\cdot\|_2$ is the l_2-norm. This problem can be obtained by using singular value threshold operator, and we get

$$\begin{aligned} Z_{k+1} &= \Phi_{\frac{1}{\eta_Z\beta}}(Z_k + \frac{1}{\eta_Z}(Y^T(Y - YZ_k - E_k + M_{1\,k}/\mu) \\ &\quad -(Z_k - J_k + M_{2\,k}/\beta)) \end{aligned} \tag{22}$$

where Φ is SVT operator [35] and $\frac{1}{\eta_Z\beta}$ is the threshold of Φ. Eq. (19) can be solved directly by the expansion operator,

$$E_{k+1} = \Psi_{\frac{\lambda}{\mu}}(Y - YZ_k + M_{1\,k}/\mu) \tag{23}$$

where Ψ is expansion operator, and $\frac{\lambda}{\mu}$ is the threshold of the expansion operator. Eq. (20) has independent solutions for different columns, so it can be decomposed into n independent subproblems as follows,

$$\begin{aligned} [J_{k+1}]_{*,i} &= \arg\min_{[J]_{*,i}\geq 0} \beta[T]_{*,i}^T[J]_{*,i} + \frac{\mu}{2}\left\|[Z_k]_{*,i} - [J]_{*,i} + [M_{2\,k}]_{*,i}/\mu\right\|_2^2 \\ &= \max\left([Z_k]_{*,i} + [M_{2\,k}]_{*,i}/\mu - \beta[T]_{*,i}/\mu, 0\right)\ (i = 1, \ldots, n) \end{aligned} \tag{24}$$

where $[\cdot]_{*,i}$ is the i-th column of a matrix and $\|\cdot\|_2$ donates the l_2-norm. Other parameters are iterated by using the formula,

$$M_{1\,k+1} = M_{1\,k} + \mu_k(Y - YZ_{k+1} - E_{k+1}) \tag{25}$$

$$M_{2\,k+1} = M_{2\,k} + \mu_k(Z_{k+1} - J_{k+1}) \tag{26}$$

$$\mu_{k+1} = \min(\mu_{\max}, \rho\mu_k) \tag{27}$$

4.2. Solving HWLRR

In our HWLRR algorithm, a hierarchical weighted matrix is defined to find the data points that may belong to the same subspace. This hierarchical weighted matrix is constructed by affinity propagation. Introducing this matrix to the original LRR, the new model can be solved by the enhanced Lagrange function.

The HWLRR algorithm can be described as follows.

1. Construct the first order adjacency matrix W by the k-nearest neighbors of each sample.
2. Figure out the completely reachable matrix P_s.
3. Obtain the hierarchical weighted matrix T to represent the penalty of each point.
4. Add matrix T to the original LRR model as Eq. (13) and solve the model using Algorithm 1.

4.3. Complexity analysis

In our HWLRR, the computational cost includes two parts, affinity propagation and solving the objective function. For a given dataset $Y = [y_1, y_2, \cdots, y_n]$, each data y_i has d dimensions and k nearest neighbors. The computational complexity of affinity propagation can be obtained from Eq. (9). Because Eq. (9) is a recurrence formula, its computational complexity is $O(t_1(n^2 + 5n))$, where t_1 is the number of iterations and $t_1 \leq log_k n$. If r is the lowest rank learned and t_2 is the number of total iterations, the computational cost of Eq. (22) is $O(rn^2)$. Meanwhile, in Eqs. (23) and (24), the complexity of updating the matrixes is $O(n^2) + O(n)$. Thus, the computational complexity of our algorithm is $O(t_1(n^2 + 5n) + t_2(rn^2 + n^2 + n))$.

Algorithm 1 Solving HWLRR.

Input: dataset Y, hierarchical weighted T, parameter $\lambda > 0$, $\beta > 0$
Output: optimal solutions Z and E
Initialize: $Z = J = M_2 = 0$, $E = M_1 = 0$, $\mu_0 = 1$, $u_{\max} = 10000$, $\eta_Z = 2\|Y\|_2$, $\xi = 0.00001$, $\rho = 1.01$, $t = 0$
While not converged **do**
1: Update Z by $Z_{k+1} = \Phi_{\frac{1}{\eta_Z\beta}}(Z_k + \frac{1}{\eta_Z}(Y^T(Y - YZ_k - E_k + M_{1\,k}/\mu) - (Z_k - J_k + M_{2\,k}/\beta)))$
2: Update E by $E_{k+1} = \Psi_{\frac{\lambda}{\mu}}(Y - YZ_k + M_{1\,k}/\mu)$
3: Update J by $\left[J_{k+1}\right]_{*,i} = \max([Z_k]_{*,i} + [M_{2\,k}]_{*,i}/\mu - \beta[T]_{*,i}/\mu, 0)(i = 1, \ldots, n)$
4: Update M_1 by $M_{1\,k+1} = M_{1\,k} + \mu_k\left(Y - YZ_{k+1} - E_{k+1}\right)$
5: Update M_2 by $M_{2\,k+1} = M_{2\,k} + \mu_k(Z_{k+1} - J_{k+1})$
6: Update μ by $\mu_{k+1} = \min(\mu_{\max}, \rho\mu_k)$
7: t = t + 1
8: check the convergence conditions $\|Z_{k+1} - Z_k\|/\|Z_k\| \leq \xi$
End while

5. Experiments

In this section, the effectiveness of HWLRR algorithm is verified through the experiments. Here, our method is tested in clustering and semi-supervised classification respectively. In the experiments, six datasets, including COIL20, UMIST, ORL, MNIST, C-Cube and PenDigits are used. All experiments are run on MATLAB 2019b on a PC with AMD(R) Ryzen R7-3700X CPU at 4.2 GHz, RAM 16.00GB, and Windows 10 operating system.

5.1. Datasets

- COIL20[1] [36] is a dataset proposed by Columbia University. There are 20 objects in this dataset, and each of them is photographed 72 images with different angles. For computational efficiency, we turn the images from 128×128 pixels to 32×32 pixels and transform them into gray-scale so that they can be converted as 1024 dimensional vectors.
- ORL dataset[2] [37] contains 10 humans' frontal face images with size of 32×32 pixels. Each person is photographed with 40 images that are taken varying the facial expressions, lighting and facial details.
- UMIST[3] [38] contains 564 images of 20 subjects, and the size of each image is 23×28 pixels. For each subject, it consists of images varying in race, sex or appearance and contains a range of poses from profile to frontal views. The images belong to the same person are numbered chronologically.
- MNIST[4] [39] is one of the most famous datasets in machine learning. It contains 70,000 handwritten images of 10 numbers and each image is 28×28 pixels. In the experiments, we randomly select 200 images from each number to make up a new dataset with 2000 digits.

[1] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.
[2] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
[3] https://www.sheffield.ac.uk/eee/research/iel/research/face.
[4] http://yann.lecun.com/exdb/mnist/.

Table 1
Clustering performance metrics (average ± std) and running times of different methods on COIL20 database. Bold numbers denote the best results.

Methods	ARI	NMI	ACC	Precision	F-score	Time(*s*)
K-means+	0.497±0.036	0.718±0.017	0.545±0.047	0.614±0.023	0.525±0.034	0.12
SC	0.572±0.019	0.759±0.008	0.635±0.018	0.618±0.019	0.594±0.018	12.93
LRR	0.414±0.024	0.663±0.012	0.514±0.026	0.485±0.017	0.445±0.022	74.96
LatLRR	0.184±0.012	0.452±0.012	0.277±0.014	0.307±0.024	0.234±0.011	221.82
NSLLRR	0.497±0.018	0.707±0.011	0.607±0.017	0.542±0.015	0.522±0.017	3221.95
LRRADP	0.621±0.031	0.792±0.013	0.678±0.039	0.673±0.019	0.640±0.029	1258.43
AWNLRR	0.693±0.017	0.838±0.006	0.740±0.016	0.730±0.011	0.708±0.016	68.91
N2D	0.710±0.013	0.846±0.006	0.764±0.012	0.749±0.009	0.725±0.010	608.91
HWLRR	**0.778±0.031**	**0.908±0.011**	**0.809±0.034**	**0.845±0.016**	**0.790±0.029**	196.70

Table 2
Clustering performance metrics (average ± std) and running times of different methods on ORL database. Bold numbers denote the best results.

Methods	ARI	NMI	ACC	Precision	F-score	Time(*s*)
K-means+	0.309±0.034	0.713±0.018	0.480±0.036	0.438±0.028	0.328±0.033	0.03
SC	0.509±0.029	0.805±0.013	0.631±0.026	0.562±0.027	0.520±0.028	1.20
LRR	0.535±0.025	0.818±0.010	0.650±0.020	0.594±0.022	0.546±0.024	8.94
LatLRR	0.529±0.031	0.817±0.016	0.656±0.026	0.584±0.028	0.541±0.030	29.10
NSLLRR	0.538±0.021	0.822±0.007	0.653±0.022	0.611±0.017	0.550±0.020	126.70
LRRADP	0.460±0.022	0.781±0.008	0.590±0.022	0.519±0.018	0.473±0.021	87.67
AWNLRR	0.490±0.022	0.811±0.006	0.637±0.023	0.601±0.016	0.504±0.021	4.95
N2D	0.388±0.012	0.748±0.004	0.523±0.013	0.446±0.008	0.404±0.009	186.95
HWLRR	**0.571±0.012**	**0.834±0.004**	**0.679±0.025**	**0.632±0.008**	**0.581±0.012**	11.05

Table 3
Clustering performance metrics (average ± std) and running times of different methods on UMIST database. Bold numbers denote the best results.

Methods	ARI	NMI	ACC	Precision	F-score	Time(*s*)
K-means+	0.284±0.017	0.595±0.012	0.403±0.021	0.341±0.013	0.323±0.016	0.02
SC	0.341±0.020	0.635±0.016	0.449±0.015	0.380±0.019	0.375±0.019	1.80
LRR	0.361±0.035	0.651±0.021	0.473±0.025	0.402±0.031	0.394±0.033	14.32
LatLRR	0.145±0.014	0.424±0.018	0.320±0.017	0.237±0.022	0.196±0.012	43.04
NSLLRR	0.426±0.032	0.698±0.016	0.536±0.026	0.463±0.035	0.456±0.030	268.65
LRRADP	0.332±0.032	0.626±0.017	0.433±0.029	0.378±0.032	0.367±0.031	149.84
AWNLRR	0.360±0.026	0.646±0.019	0.462±0.024	0.399±0.025	0.393±0.024	9.06
N2D	0.477±0.011	0.740±0.006	0.562±0.008	0.517±0.010	0.505±0.009	249.06
HWLRR	**0.521±0.030**	**0.756±0.012**	**0.604±0.028**	**0.579±0.022**	**0.547±0.028**	27.58

Table 4
Clustering performance metrics (average ± std) and running times of different methods on MNIST database. Bold numbers denote the best results.

Methods	ARI	NMI	ACC	Precision	F-score	Time(*s*)
K-means+	0.353±0.022	0.493±0.014	0.518±0.029	0.442±0.019	0.421±0.019	0.30
SC	0.379±0.020	0.493±0.016	0.566±0.029	0.451±0.017	0.442±0.018	36.32
LRR	0.022±0.002	0.055±0.001	0.166±0.006	0.129±0.002	0.123±0.001	59.13
LatLRR	0.024±0.002	0.048±0.002	0.181±0.006	0.138±0.008	0.128±0.004	132.79
NSLLRR	0.328±0.019	0.452±0.012	0.517±0.022	0.404±0.014	0.396±0.016	4732.76
LRRADP	0.427±0.018	0.573±0.013	0.591±0.034	0.536±0.011	0.489±0.015	3725.37
AWNLRR	0.493±0.031	0.624±0.017	0.629±0.045	0.580±0.034	0.547±0.028	171.46
N2D	**0.629±0.012**	**0.711±0.006**	**0.749±0.015**	**0.715±0.011**	**0.669±0.009**	745.46
HWLRR	0.577±0.028	0.694±0.017	0.735±0.042	0.669±0.027	0.622±0.025	511.42

- C-Cube dataset[5] [40] is a cursive word dataset with 50,000 samples. It includes both upper and lower case of 26 letters. We randomly select 40 samples of each class to make up a new dataset with 2000 digits.
- PenDigits dataset[6] [41] is a handwritten images for 10 numbers. It was created as a digit database by collecting 250 samples from 44 writers. In the experiments, we randomly select 100 images from each number and make up a new dataset with 1000 digits.

[5] http://ccc.idiap.ch.

[6] http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits.

5.2. Image clustering

To extensively assess the clustering performance of our approach, we compare it with the following methods:

- K-means++ clustering algorithm (K-means++) [42]
- Spectral clustering (SC) [23]
- Low-rank representation (LRR) [43]
- Latent low-rank representation (LatLRR) [44]

Table 5
Clustering performance metrics (average ± std) and running times of different methods on C-Cube database. Bold numbers denote the best results.

Methods	ARI	NMI	ACC	Precision	F-score	Time(*s*)
K-means+	0.218±0.006	0.544±0.005	0.346±0.015	0.263±0.008	0.235±0.006	0.04
SC	0.233±0.011	0.556±0.007	0.383±0.014	0.255±0.012	0.248±0.011	15.70
LRR	0.201±0.012	0.515±0.011	0.349±0.018	0.221±0.012	0.217±0.012	1.61
LatLRR	0.200±0.007	0.517±0.007	0.348±0.013	0.222±0.007	0.217±0.007	0.69
NSLLRR	0.230±0.005	0.562±0.003	0.381±0.010	0.257±0.006	0.246±0.005	3967.60
LRRADP	0.229±0.005	0.553±0.006	0.375±0.011	0.250±0.005	0.245±0.005	3224.28
AWNLRR	0.230±0.008	0.553±0.006	0.385±0.009	0.257±0.007	0.246±0.007	149.83
N2D	0.279±0.007	0.583±0.007	0.412±0.008	0.307±0.007	0.283±0.007	576.83
HWLRR	**0.283±0.008**	**0.600±0.005**	**0.437±0.008**	**0.317±0.006**	**0.298±0.008**	423.59

Table 6
Clustering performance metrics (average ± std) and running times of different methods on PenDigits database. Bold numbers denote the best results.

Methods	ARI	NMI	ACC	Precision	F-score	Time(*s*)
K-means+	0.488±0.039	0.661±0.018	0.633±0.053	0.596±0.022	0.543±0.033	0.01
SC	0.516±0.020	0.667±0.006	0.651±0.037	0.591±0.017	0.566±0.017	7.87
LRR	0.515±0.020	0.656±0.015	0.671±0.043	0.579±0.015	0.564±0.017	0.46
LatLRR	0.378±0.019	0.530±0.018	0.567±0.022	0.454±0.020	0.441±0.018	0.30
NSLLRR	0.459±0.029	0.606±0.016	0.620±0.035	0.526±0.021	0.514±0.025	377.76
LRRADP	0.528±0.023	0.664±0.013	0.675±0.035	0.610±0.007	0.574±0.020	239.80
AWNLRR	0.525±0.025	0.659±0.016	0.679±0.023	0.588±0.020	0.573±0.023	26.70
N2D	**0.657±0.012**	**0.794±0.008**	0.766±0.011	**0.746±0.010**	**0.693±0.012**	276.70
HWLRR	0.621±0.020	0.761±0.012	**0.770±0.027**	0.700±0.012	0.661±0.017	35.12

Table 7
Classification performance metrics (average ± std) on COIL20 database. #Tr denotes the number of labeled samples of a subject. Bold numbers denote the best results.

#Tr	Metrics	LRR	LatLRR	NSLLRR	LRRADP	AWNLRR	HWLRR
1	ACC	0.362±0.130	0.119±0.024	0.226±0.045	0.662±0.032	0.774±0.019	**0.861±0.013**
	Recall	0.173±0.088	0.057±0.115	0.094±0.063	0.514±0.023	0.655±0.023	**0.699±0.015**
	F-score	0.246±0.074	0.094±0.004	0.150±0.024	0.563±0.031	0.711±0.019	**0.794±0.029**
2	ACC	0.420±0.100	0.126±0.022	0.222±0.064	0.748±0.019	0.805±0.012	**0.865±0.008**
	Recall	0.186±0.081	0.058±0.085	0.103±0.099	0.629±0.018	0.710±0.019	**0.691±0.015**
	F-score	0.264±0.093	0.096±0.005	0.158±0.031	0.660±0.019	0.747±0.015	**0.794±0.012**
3	ACC	0.606±0.077	0.134±0.027	0.222±0.046	0.781±0.014	0.829±0.018	**0.885±0.013**
	Recall	0.360±0.047	0.061±0.062	0.102±0.090	0.674±0.016	0.725±0.010	**0.741±0.023**
	F-score	0.428±0.112	0.099±0.007	0.156±0.029	0.701±0.015	0.767±0.026	**0.830±0.031**
4	ACC	0.587±0.223	0.124±0.033	0.250±0.057	0.806±0.015	0.840±0.016	**0.890±0.011**
	Recall	0.412±0.060	0.059±0.127	0.114±0.077	0.707±0.017	0.740±0.021	**0.756±0.012**
	F-score	0.461±0.194	0.097±0.005	0.169±0.035	0.728±0.019	0.779±0.020	**0.842±0.016**
5	ACC	0.758±0.062	0.139±0.033	0.270±0.070	0.826±0.015	0.848±0.008	**0.887±0.012**
	Recall	0.615±0.058	0.063±0.126	0.137±0.087	0.731±0.019	0.748±0.026	**0.744±0.014**
	F-score	0.641±0.089	0.099±0.008	0.189±0.046	0.753±0.019	0.791±0.010	**0.832±0.023**
6	ACC	0.793±0.021	0.161±0.034	0.260±0.037	0.829±0.014	0.849±0.006	**0.903±0.007**
	Recall	0.675±0.026	0.067±0.070	0.120±0.046	0.737±0.016	0.753±0.020	**0.783±0.015**
	F-score	0.693±0.028	0.103±0.010	0.175±0.030	0.757±0.020	0.791±0.011	**0.861±0.012**
7	ACC	0.821±0.022	0.148±0.039	0.247±0.064	0.839±0.018	0.859±0.006	**0.904±0.007**
	Recall	0.707±0.024	0.062±0.091	0.117±0.074	0.757±0.021	0.774±0.016	**0.785±0.010**
	F-score	0.724±0.025	0.101±0.014	0.173±0.043	0.772±0.021	0.803±0.011	**0.864±0.010**

- Non-negative sparse laplacian regularized low-rank representation (NSLLRR) [21]
- Low-rank representation with adaptive distance penalty (LRRADP) [25]
- Adaptive weighted nonnegative low-rank representation (AWNLRR) [22]
- N2D [45]

K-means++ is an advanced K-means clustering algorithm and is regarded as a baseline for clustering. Spectral clustering (SC) is a classical clustering algorithm that clusters data by spectral decomposition. LRR, LatLRR, NSLLRR, LRRADP and AWNLRR are five representative LRR algorithms, and the representations produced by these methods are clustered by SC. N2D is a novel deep clustering method which adopts manifold learning. To compare the qualities of the final clustering results, the adjusted rand index (ARI), normalized mutual information (NMI), accuracy (ACC), precision and F-score are employed. For all the experiments, the run has been repeated ten times to eliminate the effects of randomness embedded in the algorithms. The average, standard deviation of metrics, and the time cost are given.

Table 1–6 show the clustering performance and running time of different methods on the datasets. From these tables, one can find that the proposed method obtains the best performance in most cases. It means that the hierarchical weighted matrix can capture the structure embedded in the data space better. From the comparison of K-means++, SC and LRR methods, it is obvious that low-rank representation is more effective than the original pixel feature in obtaining a better clustering performance in almost all cases. But in some datasets as COIL20, the pixel feature can obtain pretty good clustering performance compared with LRR, LatLRR and NSLLRR. From the comparison of LRR, LatLRR and

Table 8
Classification performance metrics (average ± std) on ORL database. #Tr denotes the number of labeled samples of a subject. Bold numbers denote the best results.

#Tr	Metrics	LRR	LatLRR	NSLLRR	LRRADP	AWNLRR	HWLRR
1	ACC	0.559±0.019	0.158±0.048	0.566±0.078	0.582±0.015	0.581±0.028	**0.662±0.028**
	Recall	0.287±0.022	0.038±0.110	0.305±0.028	0.359±0.013	0.222±0.037	**0.429±0.016**
	F-score	0.357±0.041	0.065±0.009	0.377±0.084	0.398±0.019	0.317±0.081	**0.506±0.033**
2	ACC	0.692±0.026	0.155±0.038	0.715±0.039	0.707±0.019	0.669±0.019	**0.745±0.023**
	Recall	0.450±0.037	0.036±0.105	0.491±0.047	0.478±0.027	0.357±0.030	**0.544±0.019**
	F-score	0.515±0.045	0.062±0.011	0.556±0.069	0.528±0.029	0.459±0.045	**0.614±0.023**
3	ACC	0.732±0.014	0.163±0.045	0.759±0.024	0.785±0.016	0.704±0.024	**0.817±0.018**
	Recall	0.497±0.018	0.035±0.044	0.559±0.022	0.574±0.031	0.406±0.028	**0.652±0.021**
	F-score	0.561±0.022	0.063±0.015	0.615±0.030	0.620±0.033	0.503±0.047	**0.699±0.023**
4	ACC	0.771±0.027	0.273±0.086	0.800±0.020	0.795±0.032	0.762±0.024	**0.847±0.024**
	Recall	0.554±0.033	0.049±0.082	0.622±0.027	0.587±0.036	0.504±0.026	**0.720±0.019**
	F-score	0.616±0.039	0.083±0.024	0.674±0.027	0.634±0.039	0.586±0.045	**0.756±0.035**
5	ACC	0.793±0.021	0.453±0.256	0.810±0.025	0.834±0.013	0.770±0.030	**0.881±0.018**
	Recall	0.568±0.019	0.159±0.120	0.626±0.027	0.658±0.020	0.503±0.032	**0.760±0.029**
	F-score	0.629±0.031	0.214±0.198	0.686±0.028	0.697±0.020	0.591±0.060	**0.788±0.027**
6	ACC	0.802±0.026	0.857±0.095	0.818±0.013	0.843±0.024	0.792±0.026	**0.895±0.016**
	Recall	0.579±0.040	0.628±0.110	0.609±0.019	0.653±0.030	0.529±0.032	**0.761±0.022**
	F-score	0.643±0.046	0.682±0.214	0.671±0.020	0.700±0.046	0.615±0.038	**0.795±0.025**
7	ACC	0.811±0.020	0.846±0.129	0.836±0.027	0.860±0.025	0.816±0.021	**0.910±0.023**
	Recall	0.567±0.029	0.558±0.154	0.621±0.046	0.659±0.029	0.549±0.029	**0.777±0.039**
	F-score	0.636±0.032	0.618±0.314	0.687±0.047	0.711±0.042	0.630±0.035	**0.811±0.044**

Table 9
Classification performance metrics (average ± std) on UMIST database. #Tr denotes the number of labeled samples of a subject. Bold numbers denote the best results.

#Tr	Metrics	LRR	LatLRR	NSLLRR	LRRADP	AWNLRR	HWLRR
1	ACC	0.444±0.030	0.133±0.017	0.234±0.060	0.462±0.020	0.103±0.025	**0.794±0.017**
	Recall	0.266±0.031	0.056±0.097	0.101±0.115	0.284±0.043	0.056±0.123	**0.677±0.022**
	F-score	0.295±0.028	0.087±0.005	0.160±0.024	0.319±0.031	0.106±0.014	**0.727±0.021**
2	ACC	0.620±0.049	0.136±0.013	0.300±0.072	0.577±0.020	0.094±0.014	**0.853±0.017**
	Recall	0.448±0.052	0.056±0.102	0.135±0.081	0.422±0.038	0.053±0.133	**0.790±0.017**
	F-score	0.472±0.057	0.087±0.005	0.195±0.044	0.446±0.037	0.100±0.004	**0.813±0.022**
3	ACC	0.684±0.048	0.132±0.014	0.333±0.094	0.651±0.048	0.098±0.019	**0.870±0.033**
	Recall	0.527±0.065	0.057±0.093	0.149±0.086	0.498±0.053	0.053±0.087	**0.801±0.026**
	F-score	0.549±0.066	0.092±0.004	0.216±0.058	0.524±0.054	0.101±0.005	**0.825±0.033**
4	ACC	0.761±0.034	0.147±0.019	0.457±0.102	0.718±0.043	0.113±0.024	**0.909±0.014**
	Recall	0.631±0.046	0.061±0.098	0.245±0.047	0.578±0.048	0.056±0.105	**0.843±0.014**
	F-score	0.645±0.045	0.093±0.005	0.313±0.082	0.600±0.049	0.106±0.012	**0.863±0.023**
5	ACC	0.794±0.017	0.148±0.021	0.483±0.212	0.778±0.021	0.119±0.039	**0.909±0.027**
	Recall	0.669±0.020	0.061±0.084	0.274±0.134	0.674±0.037	0.057±0.095	**0.839±0.029**
	F-score	0.685±0.024	0.094±0.006	0.343±0.202	0.691±0.039	0.107±0.019	**0.861±0.040**
6	ACC	0.842±0.029	0.152±0.022	0.770±0.120	0.782±0.041	0.116±0.018	**0.928±0.015**
	Recall	0.739±0.033	0.063±0.086	0.619±0.125	0.680±0.059	0.054±0.028	**0.863±0.014**
	F-score	0.751±0.039	0.094±0.006	0.659±0.177	0.696±0.057	0.103±0.003	**0.885±0.016**
7	ACC	0.857±0.031	0.169±0.021	0.877±0.033	0.790±0.026	0.124±0.014	**0.935±0.014**
	Recall	0.768±0.047	0.066±0.079	0.806±0.045	0.686±0.034	0.056±0.139	**0.879±0.017**
	F-score	0.775±0.044	0.099±0.004	0.820±0.045	0.706±0.036	0.106±0.006	**0.896±0.016**

NSLLRR, one can conclude that introducing manifold information (i.e., Laplacian regularization) can obtain a better performance. As the Laplacian regularization is constructed by the k-nearest neighbor relationship, this also proves the benefit of keeping the local structure of data. From all the tables, we can find that the performances of AWNLRR and LRRADP are better than NSLLRR. AWNLRR and LRRADP use the Euclidean distance as a constraint to preserve the geometric relationship among data points. One can conclude that the distance constraint is more effective than the Laplacian term in learning the intrinsic relationships of data. Both NSLLRR and HWLRR use the k-nearest neighbor relationship to learn the local structure in the data. However, from the comparison of the two methods, it is obvious that the proposed HWLRR method performs much better than NSLLRR. This is due to that high hierarchy neighbors preserve more global structure embedded in the data. Compared with N2D, our method could achieve competitive performance on three large-scale datasets and obtains better performance on the others. As can be seen, although our proposed HWLRR is not the most efficient, its computational complexity is at the same level as AWNLRR. However, solving the Eq. (24) is the bottleneck of HWLRR's computation complexity because this equation should be solved column by column.

5.3. Semi-Supervised classification

In this section, the performances of our HWLRR, LRR, LatLRR, NSLLRR, LRRADP and AWNLRR are compared through semi-supervised classification. All the algorithms compute the representation matrix firstly, then the GFHF [24] algorithm is used to predict the labels for the unlabeled points. In the experiments, the number of labeled points per cluster is varied from 1 to 7 and the labeled samples are randomly selected. For all the experiments, the run has been repeated ten times to eliminate the effects of randomness embedded in the algorithms. Here, the ACC, recall and F-score are employed to compare the qualities of the final classification results. The average and standard deviation of the metrics are given in Table 7-12.

Table 10
Classification performance metrics (average ± std) on MNIST database. #Tr denotes the number of labeled samples of a subject. Bold numbers denote the best results.

#Tr	Metrics	LRR	LatLRR	NSLLRR	LRRADP	AWNLRR	HWLRR
1	ACC	0.100±1.462	0.132±0.004	0.202±0.018	**0.322±0.097**	0.308±0.075	0.311±0.130
	Recall	0.099±0.005	0.100±0.090	0.121±0.064	0.157±0.072	0.155±0.092	**0.177±0.119**
	F-score	0.181±2.925	0.140±0.013	0.179±0.009	0.247±0.078	0.245±0.062	**0.276±0.084**
2	ACC	0.100±1.462	0.128±0.004	0.226±0.038	**0.565±0.072**	0.538±0.093	0.510±0.097
	Recall	0.099±0.005	0.100±0.118	0.138±0.093	**0.316±0.041**	0.291±0.040	0.266±0.047
	F-score	0.181±2.925	0.150±0.016	0.184±0.010	**0.391±0.092**	0.383±0.096	0.371±0.085
3	ACC	0.100±1.462	0.135±0.003	0.230±0.035	0.629±0.055	**0.683±0.059**	0.667±0.058
	Recall	0.099±0.005	0.101±0.041	0.138±0.119	0.395±0.026	**0.470±0.032**	0.429±0.045
	F-score	0.181±2.925	0.138±0.007	0.180±0.013	0.465±0.056	**0.530±0.068**	0.521±0.069
4	ACC	0.100±1.462	0.133±0.005	0.214±0.019	0.692±0.029	0.721±0.058	**0.733±0.053**
	Recall	0.099±0.005	0.100±0.108	0.128±0.078	0.497±0.020	0.527±0.030	**0.534±0.023**
	F-score	0.181±2.925	0.142±0.015	0.174±0.007	0.540±0.036	0.585±0.052	**0.600±0.057**
5	ACC	0.100±1.462	0.131±0.008	0.224±0.037	0.743±0.030	**0.774±0.027**	0.767±0.051
	Recall	0.099±0.005	0.100±0.106	0.134±0.094	0.563±0.027	**0.612±0.031**	0.600±0.036
	F-score	0.181±2.925	0.143±0.014	0.192±0.009	0.595±0.038	0.640±0.034	**0.642±0.054**
6	ACC	0.100±1.462	0.131±0.008	0.216±0.025	0.756±0.020	0.797±0.023	**0.812±0.019**
	Recall	0.099±0.005	0.100±0.104	0.128±0.068	0.583±0.021	0.640±0.024	**0.656±0.018**
	F-score	0.181±2.925	0.139±0.014	0.186±0.005	0.611±0.025	0.662±0.031	**0.685±0.027**
7	ACC	0.100±1.462	0.131±0.007	0.229±0.035	0.786±0.014	0.808±0.020	**0.813±0.037**
	Recall	0.099±0.005	0.100±0.073	0.135±0.111	0.629±0.016	0.660±0.021	**0.663±0.023**
	F-score	0.181±2.925	0.140±0.012	0.196±0.010	0.648±0.016	0.678±0.028	**0.688±0.038**

Table 11
Classification performance metrics (average ± std) on C-Cube database. #Tr denotes the number of labeled samples of a subject. Bold numbers denote the best results.

#Tr	Metrics	LRR	LatLRR	NSLLRR	LRRADP	AWNLRR	HWLRR
1	ACC	0.033±0.013	0.235±0.014	0.203±0.016	0.283±0.011	0.165±0.027	**0.369±0.019**
	Recall	0.020±0.243	0.096±0.007	0.078±0.017	0.137±0.010	0.035±0.121	**0.195±0.020**
	F-score	0.039±0.000	0.109±0.009	0.112±0.014	0.163±0.009	0.064±0.011	**0.233±0.021**
2	ACC	0.058±0.021	0.305±0.018	0.207±0.008	0.324±0.015	0.202±0.027	**0.434±0.011**
	Recall	0.020±0.203	0.137±0.015	0.080±0.023	0.152±0.011	0.051±0.076	**0.254±0.012**
	F-score	0.040±0.001	0.152±0.013	0.117±0.006	0.188±0.010	0.088±0.020	**0.286±0.014**
3	ACC	0.066±0.036	0.346±0.014	0.175±0.012	0.338±0.015	0.218±0.019	**0.465±0.011**
	Recall	0.022±0.272	0.164±0.007	0.055±0.054	0.155±0.011	0.057±0.049	**0.282±0.010**
	F-score	0.042±0.004	0.182±0.007	0.091±0.011	0.193±0.010	0.097±0.024	**0.304±0.008**
4	ACC	0.092±0.043	0.375±0.017	0.095±0.042	0.370±0.010	0.237±0.024	**0.494±0.021**
	Recall	0.023±0.268	0.178±0.011	0.034±0.136	0.170±0.008	0.061±0.039	**0.307±0.018**
	F-score	0.044±0.005	0.199±0.011	0.063±0.012	0.211±0.010	0.103±0.017	**0.327±0.020**
5	ACC	0.147±0.044	0.400±0.007	0.115±0.033	0.373±0.013	0.273±0.031	**0.509±0.011**
	Recall	0.029±0.194	0.198±0.007	0.040±0.096	0.168±0.009	0.078±0.034	**0.319±0.009**
	F-score	0.053±0.011	0.220±0.006	0.072±0.011	0.211±0.011	0.124±0.026	**0.336±0.009**
6	ACC	0.200±0.030	0.426±0.012	0.159±0.042	0.391±0.014	0.273±0.030	**0.531±0.010**
	Recall	0.033±0.126	0.214±0.009	0.053±0.082	0.180±0.010	0.080±0.040	**0.340±0.013**
	F-score	0.060±0.012	0.236±0.009	0.090±0.019	0.221±0.010	0.125±0.030	**0.357±0.012**
7	ACC	0.216±0.043	0.430±0.010	0.149±0.023	0.404±0.015	0.302±0.023	**0.542±0.005**
	Recall	0.034±0.107	0.218±0.012	0.048±0.049	0.186±0.018	0.096±0.020	**0.349±0.012**
	F-score	0.062±0.011	0.243±0.013	0.084±0.011	0.230±0.016	0.147±0.018	**0.364±0.008**

From these tables, we can see that the ACC obtained by the six algorithms increases as the number of labels increases. From comparision of NSLLRR, LRRADP and AWNLRR, it can be concluded that distance constraint is more effective than Laplacian constraint for capturing the relationship among samples. It should be pointed out that Laplacian constraint is also a distance relationship between the sample and its k-nearest neighbors because the Laplacian term is constructed by the k-nearest neighbor relationship. This can prove the efficiency of introducing more sample relationships. Furthermore, it is obvious that our HWLRR algorithm performs superior to LRR, LatLRR, NSLLRR, LRRADP and AWNLRR algorithms in most experiments. Compared with other LRR methods, it is obvious that adding the hierarchical weighted matrix can obtain a better low-rank representation matrix. For COIL20 dataset, its data structure is suitable for the hierarchical weighted matrix so that HWLRR can obtain the high classification accuracy with a few labeled data.

5.4. Effectiveness of capturing the structure

In this section, the proposed method is compared with some metric learning methods to show the effectiveness of capturing the structure. We compare our method with five state-of-the-art metric learning methods (LMNN [46], DMLMJ [47], LR-GMML [48], SRRS [49], IMLSR [9]) and a baseline Euclidean metric (EU). All methods are tested on the three datasets, i.e. COIL100 [50], ORL and USPS [51]. To be fair, the number of labeled samples and the classification method are the same as Cai et al. [9]. All the experiments are repeated 20 times and the recognition rate is used for comparison. It should be pointed out that the performance of the five metric learning methods are excerpted from [9].

As shown in Table 13, it can be found that our method obtains the best performance. Moreover, all metric learning methods perform better than EU because metric learning methods can capture the structure in the data space. Preserving the manifold struc-

Table 12
Classification performance metrics (average ± std) on PenDigits database. #Tr denotes the number of labeled samples of a subject. Bold numbers denote the best results.

#Tr	Metrics	LRR	LatLRR	NSLLRR	LRRADP	AWNLRR	HWLRR
1	ACC	0.371±0.021	0.276±0.044	0.334±0.039	**0.629±0.048**	0.275±0.107	0.594±0.061
	Recall	0.226±0.035	0.153±0.092	0.181±0.035	**0.446±0.041**	0.147±0.073	0.417±0.038
	F-score	0.284±0.022	0.215±0.017	0.247±0.024	0.503±0.046	0.247±0.056	**0.513±0.038**
2	ACC	0.297±0.051	0.357±0.041	0.329±0.052	**0.696±0.045**	0.418±0.147	0.667±0.026
	Recall	0.170±0.106	0.201±0.044	0.194±0.048	**0.512±0.038**	0.230±0.115	0.465±0.016
	F-score	0.252±0.018	0.254±0.026	0.259±0.036	**0.570±0.039**	0.334±0.119	0.563±0.028
3	ACC	0.352±0.054	0.360±0.082	0.298±0.069	0.715±0.036	0.412±0.134	**0.720±0.054**
	Recall	0.198±0.052	0.211±0.074	0.163±0.060	0.504±0.029	0.218±0.087	**0.508±0.030**
	F-score	0.277±0.026	0.265±0.053	0.244±0.030	0.592±0.032	0.319±0.118	**0.595±0.061**
4	ACC	0.373±0.087	0.358±0.061	0.348±0.090	0.745±0.034	0.587±0.100	**0.752±0.061**
	Recall	0.215±0.096	0.186±0.084	0.202±0.069	0.553±0.034	0.382±0.076	**0.563±0.034**
	F-score	0.287±0.037	0.252±0.032	0.276±0.044	0.627±0.038	0.472±0.103	**0.631±0.072**
5	ACC	0.496±0.076	0.364±0.058	0.428±0.052	0.771±0.018	0.593±0.072	**0.785±0.049**
	Recall	0.293±0.044	0.179±0.076	0.263±0.021	0.610±0.017	0.348±0.044	**0.620±0.026**
	F-score	0.358±0.051	0.254±0.028	0.326±0.038	0.648±0.016	0.458±0.082	**0.674±0.055**
6	ACC	0.515±0.055	0.393±0.118	0.466±0.027	0.764±0.019	0.645±0.062	**0.794±0.049**
	Recall	0.293±0.051	0.193±0.033	0.304±0.035	0.607±0.018	0.420±0.023	**0.620±0.021**
	F-score	0.367±0.057	0.271±0.064	0.358±0.021	0.647±0.021	0.512±0.081	**0.683±0.057**
7	ACC	0.323±0.099	0.524±0.130	0.450±0.062	0.785±0.018	0.665±0.069	**0.833±0.050**
	Recall	0.164±0.063	0.281±0.037	0.306±0.034	0.627±0.020	0.452±0.027	**0.686±0.031**
	F-score	0.249±0.045	0.353±0.105	0.363±0.047	0.660±0.026	0.536±0.082	**0.730±0.058**

Table 13
The recognition rates (average ± std) on three datasets. Bold numbers denote the best results.

Method	COIL100	ORL	USPS
EU	89.82±0.86	76.80±2.49	89.83±0.86
LMNN	91.42±0.98	85.02±1.68	91.34±1.13
DMLMJ	91.98±1.35	86.30±2.74	85.35±1.30
LR-GMML	89.83±1.10	87.80±1.71	91.06±0.99
SRRS	91.18±1.09	87.31±1.77	90.46±0.84
IMLSR	92.45±1.10	89.43±1.85	91.59±0.89
HWLRR	**94.03±0.97**	**91.58±1.55**	**92.14±0.75**

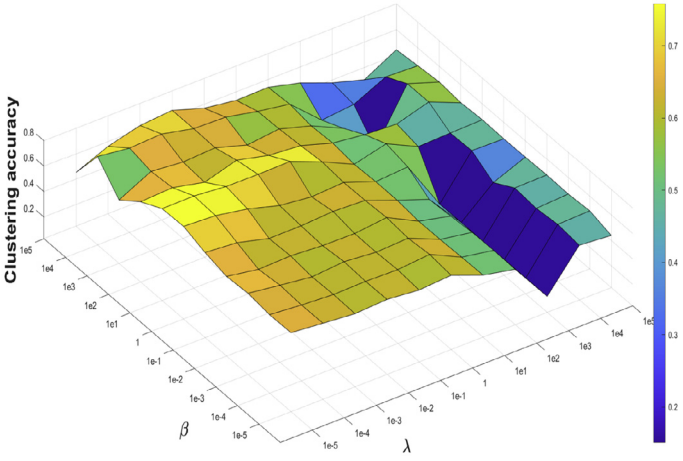

Fig. 7. The influence of λ and β on our HWLRR algorithm.

ture in data can improve the classification performance [52], thus IMLSR is the best metric learning method in the experiments.

5.5. Effect of the parameters

There are two parameters affecting the performance of our HWLRR. In the following, we study the influence of λ and β on the accuracy by varying the value of λ and β on UMIST dataset. Here, we vary one parameter and keep the other one fixed. The parameters increase from 10^{-5} to 10^{5}. The results are showed in Fig. 7. In the figure, the Z-axis shows the ACC obtained with different parameter settings. From the results, we find that the effect of λ and β is not remarkable in $[10^{-5}, 1]$. Therefore, we set $\lambda = \beta = 0.1$ in our experiments.

6. Conclusion

In this paper, a novel hierarchical weighted low-rank representation is proposed to capture the high hierarchy relationship among data points better by hierarchical weight. In HWLRR algorithm, a hierarchical weighted matrix is defined firstly. In this matrix, the weights of samples depend on the hierarchy that the samples lie on. Explained by probability, the samples obtain the same weights if they lie on the same hierarchy, and get different weights when they lie on different hierarchies. With the hierarchical weighted matrix, HWLRR can learn local neighbor relationships by k-nearest neighbor and also preserve the global clustering structure by the higher hierarchy neighbors. Therefore, HWLRR is suitable for semi-supervised label propagation methods and clustering methods (e.g., GFHF and SC). All the experimental results described in this paper show the effectiveness of our method.

Although the results presented here are extremely encouraging, there is an issue that deserves in-depth study in the future. The affinity propagation used here only stops when no sample can be found. Therefore, some samples in the different subspaces may be connected and this will deteriorate the clustering performance of the algorithm. A mechanism that allows the propagation to stop automatically should be investigated.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the editors and the anonymous referees for their helpful comments to improve the quality of the paper. This work was supported by the National Key Research and Development of China (no. 2018AAA0102100), the National Natural Science Foundation of China (nos. 61532005, U1936212), and the Fundamental Research Funds for the Central Universities (no. 2019YJS051).

References

[1] 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.
[2] 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.
[3] Z. He, L. Liu, S. Zhou, Y. Shen, Learning group-based sparse and low-rank representation for hyperspectral image classification, Pattern Recognit. 60 (2016) 1041–1056.
[4] Z. Zhang, K. Zhao, Low-rank matrix approximation with manifold regularization, IEEE Trans. Pattern Anal. Mach. Intell. 35 (7) (2012) 1717–1729.
[5] H. Du, X. Zhang, Q. Hu, Y. Hou, Sparse representation-based robust face recognition by graph regularized low-rank sparse representation recovery, Neurocomputing 164 (2015) 220–229.
[6] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, S. Zhan, Low-rank preserving projection via graph regularized reconstruction, IEEE Trans. Cybernet. 49 (4) (2018) 1279–1291.
[7] C. Chen, C. Wei, Y.F. Wang, Low-rank matrix recovery with structural incoherence for robust face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2618–2625.
[8] M. Zhang, N. Wang, Y. Li, X. Gao, Deep latent low-rank representation for face sketch synthesis, IEEE Trans. Neural Netw. Learn. Syst. 30 (10) (2019) 3109–3123.
[9] L. Cai, S. Ying, Y. Peng, C. He, S. Du, Intrinsic metric learning with subspace representation, IEEE Access 7 (2019) 68572–68583.
[10] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 1–37.
[11] J. Chen, J. Yang, Robust subspace segmentation via low-rank representation, IEEE Trans. Cybern. 44 (8) (2014) 1432–1445.
[12] H. Zhang, Z. Lin, C. Zhang, J. Gao, Robust latent low rank representation for subspace clustering, Neurocomputing 145 (2014) 369–373.
[13] R. Liu, Z. Lin, F. De la Torre, Z. Su, Fixed-rank representation for unsupervised visual learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 598–605.
[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) (2016) 1828–1838.
[15] X. Fang, Y. Xu, X. Li, Z. Lai, W.K. Wong, Learning a nonnegative sparse graph for linear regression, IEEE Trans. Image Process. 24 (9) (2015) 2760–2771.
[16] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[17] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[18] X. He, D. Cai, S. Yan, H. Zhang, Neighborhood preserving embedding, in: Proceedings of the IEEE International Conference on Computer Vision, 2005, pp. 1208–1213.
[19] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, D. Cai, Graph regularized sparse coding for image representation, IEEE Trans. Image Process. 20 (5) (2010) 1327–1336.
[20] X. Lu, Y. Wang, Y. Yuan, Graph-regularized low-rank representation for destriping of hyperspectral images, IEEE Trans. Geosci. Remote Sens. 51 (7) (2013) 4009–4018.
[21] M. Yin, J. Gao, Z. Lin, Laplacian regularized low-rank representation and its applications, IEEE Trans. Pattern Anal. Machine Intell. 38 (3) (2016) 504–517.
[22] J. Wen, B. Zhang, Y. Xu, J. Yang, N. Han, Adaptive weighted nonnegative low-rank representation, Pattern Recognit. 81 (2018) 326–340.
[23] X.S. Yu, J. Shi, Multiclass spectral clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2003, pp. 313–319.
[24] X. Zhu, Z. Ghahramani, J.D. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the International Conference on Machine Learning, 2003, pp. 912–919.
[25] 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.
[26] 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. Intell. 35 (1) (2012) 171–184.
[27] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Proceedings of the International Conference on Neural Information Processing Systems, 2002, pp. 585–591.
[28] K. Fukunaga, L. Hostetler, K-nearest-neighbor bayes-risk estimation, IEEE Trans. Inf. Theory 21 (3) (1975) 285–293.
[29] X. Wang, A fast exact k-nearest neighbors algorithm for high dimensional search using k-means clustering and triangle inequality, in: Proceedings of the International Joint Conference on Neural Networks, 2011, pp. 1293–1299.
[30] P.A. Gagniuc, Markov Chains: From Theory to Implementation and Experimentation, John Wiley & Sons, 2017.
[31] L. Zhuang, Z. Zhou, S. Gao, J. Yin, Z. Lin, Y. Ma, Label information guided graph construction for semi-supervised learning, IEEE Trans. Image Process. 26 (9) (2017) 4182–4192.
[32] F. Camastra, A. Staiano, Intrinsic dimension estimation: advances and open problems, Inf. Sci. 328 (2016) 26–41.
[33] 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.
[34] J. Yang, X. Yuan, Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization, Math. Comput. 82 (281) (2013) 301–329.
[35] J. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2010) 1956–1982.
[36] S.A. Nene, S.K. Nayar, H. Murase, et al., Columbia object image library (coil-20), Tech. Report CUCS-005-96 (1996) 1–6.
[37] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: IEEE workshop on applications of computer vision, 1994, pp. 138–142.
[38] D.B. Graham, N. Allinson, Face recognition: from theory to applications, NATO ASI Ser. F. Comput. Syst. Sci. 163 (1998) 446–456.
[39] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[40] F. Camastra, M. Spinetti, A. Vinciarelli, Offline cursive character challenge: a new benchmark for machine learning and pattern recognition algorithms., in: Proceedings of the International Conference on Pattern Recognition, 2006, pp. 913–916.
[41] F. Alimoglu, E. Alpaydin, Y. Denizhan, Combining multiple classifiers for pen-based handwritten digit recognition, Doc. Anal. Recognit. 3 (1996) 637–640.
[42] D. Arthur, S. Vassilvitskii, K-means++: The advantages of careful seeding, in: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 2007, pp. 1027–1035.
[43] 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. Intell. 35 (1) (2012) 171–184.
[44] 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.
[45] R. McConville, R. Santos-Rodriguez, R.J. Piechocki, I. Craddock, N2d:(not too) deep clustering via clustering the local manifold of an autoencoded embedding, 2019, arXiv preprint arXiv:1908.05968.
[46] K.Q. Weinberger, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2009) 207–244.
[47] B. Nguyen, C. Morell, B.D. Baets, Supervised distance metric learning through maximization of the jeffrey divergence, Pattern Recognit. 64 (2017) 215–225.
[48] M. Bhutani, P. Jawanpuria, H. Kasai, B. Mishra, Low-rank geometric mean metric learning, in: Proceedings of the International Conference on Machine Learning, 2018, pp. 1–4.
[49] S. Li, Y. Fu, Learning robust and discriminative subspace with low-rank constraints, IEEE Trans. Neural Netw. Learn. Syst. 27 (11) (2016) 2160–2173.
[50] S.A. Nene, S.K. Nayar, H. Murase, et al., Columbia object image library (coil-100), Tech. Report CUCS-006-96 (1996).
[51] J.J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 550–554.
[52] S. Ying, Z. Wen, J. Shi, Y. Peng, J. Peng, H. Qiao, Manifold preserving: an intrinsic approach for semisupervised distance metric learning, IEEE Trans. Neural Netw. Learn. Syst. 29 (7) (2018) 2731–2742.

Zhiqiang Fu received the B.S. degree in Beijing Jiaotong University in 2017. He is currently pursuing the Ph.D. degree in Institute of Information Science, Beijing Jiaotong University. His current research interests include pattern recognition and clustering.

Yao Zhao received the B.S. degree from Fuzhou Unive rsity, Fuzhou, China, in 1989, and the M.E. degree from Southeast University, Nanjing, China, in 1992, both from the Radio Engineering Department, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1996. He became an Associate Professor at BJTU in 1998 and became a Professor in 2001. In October 2015, he visited the Swiss Federal Institute of Technology, Lausanne, Switzerland (EPFL). From December 2017 to March 2018, he visited the Universit y of Southern California, Los Angeles, USA. He is currently the Director of the Institute of Information Science, BJTU. His current research interests include image/video coding, digital watermarking and forensics, video analysis and understanding and arti ficial intelligence. Dr. Zhao serves on the Editorial Boards of several international journals, including as an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS, a Senior Associate Editor of the IEEE SIGNAL PROCESSING LETTERS, and an Area Editor of Signal Processing: Image Communication. He was named a Distinguished Young Scholar by the National Science Foundation of China in 2010, and was elected as a Chang Jiang Scholar of Ministry of Education of China in 2013. He is a Fellow of the IET.

Dongxia Chang received the M.S. degree in mathematics from Xidian University and the Ph.D. degree in control science and engineering from Tsinghua University in 2003 and 2009, respectively. She is currently an associate professor of the Institute of Information Science of Beijing Jiaotong University. Her research interests include clustering, pattern recognition, and image segmentation.

Yiming Wang received the B.Eng degree in computer science and technology from the Shandong Normal University, Jinan, China, in 2017, and the MA.Eng degree in electronics and communications engineering from Beijing Jiaotong University, Beijing, China, in 2019. He is currently working toward the Ph.D. degree in the school of computer and information technology at Beijing Jiaotong University. His current research interests include clustering analysis and deep learning.