One-step Low-Rank Representation for Clustering

Zhiqiang Fu
zhiqiangfu@bjtu.edu.cn
[1]Institute of Information Science, Beijing Jiaotong University
[2]Beijing Key Laboratory of Advanced Information Science and Network Technology
Beijing, China

Yao Zhao*
yzhao@bjtu.edu.cn
[1]Institute of Information Science, Beijing Jiaotong University
[2]Beijing Key Laboratory of Advanced Information Science and Network Technology
Beijing, China

Dongxia Chang
dxchang@bjtu.edu.cn
[1]Institute of Information Science, Beijing Jiaotong University
[2]Beijing Key Laboratory of Advanced Information Science and Network Technology
Beijing, China

Yiming Wang
wangym@bjtu.edu.cn
Institute of Information Science, Beijing Jiaotong University
Beijing, China

Jie Wen
jiewen_pr@126.com
Shenzhen Key Laboratory of Visual Object Detection and Recognition, Harbin Institute of Technology
Shenzhen, China

Xingxing Zhang
xxzhang2020@mail.tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing, China

Guodong Guo
guoguodong01@baidu.com
Baidu Research
China

ABSTRACT

Existing low-rank representation-based methods adopt a two-step framework, which must employ an extra clustering method to gain labels after representation learning. In this paper, a novel one-step representation-based method, i.e., One-step Low-Rank Representation (OLRR), is proposed to capture multi-subspace structures for clustering. OLRR integrates the low-rank representation model and clustering into a unified framework. Thus it can jointly learn the low-rank subspace structure embedded in the database and gain the clustering results. In particular, by approximating the representation matrix with two same clustering indicator matrices, OLRR can directly show the probability of samples belonging to each cluster. Further, a probability penalty is introduced to ensure that the samples with smaller distances are more inclined to be in the same cluster, thus enhancing the discrimination of the clustering indicator matrix and resulting in a more favorable clustering performance. Moreover, to enhance the robustness against noise, OLRR uses the probability to guide denoising and then performs representation learning and clustering in a recovered clean space. Extensive experiments well demonstrate the robustness and effectiveness of OLRR. Our code is publicly available at:https://github.com/fuzhiqiang1230/OLRR.

*Corresponding author.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548293

CCS CONCEPTS

• **Computing methodologies** → **Cluster analysis**; • **Information systems** → **Clustering**.

KEYWORDS

Low-rank representation, data clustering, affinity matrix, subspace learning

ACM Reference Format:
Zhiqiang Fu, Yao Zhao, Dongxia Chang, Yiming Wang, Jie Wen, Xingxing Zhang, Guodong Guo. 2022. One-step Low-Rank Representation for Clustering. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548293

1 INTRODUCTION

Clustering is one of the main research of unsupervised learning and can automatically divide the given samples into different clusters with little or no prior knowledge [33, 34, 37, 38]. Since clustering can save a huge mount cost of manually labeled databases, it has been widely used in many areas, like image processing [6, 41], knowledge discovery [16] and data mining [23, 30]. To meet mission requirements, many clustering methods are proposed as graph clustering [20], weighted K-means [23], DBSCAN [7]. These clustering methods can generally be classified into two groups, i.e., non-fuzzy clustering and fuzzy clustering. Non-fuzzy clustering (also named hard clustering) groups the samples so that each sample is only assigned to one cluster [42]. Different from hard clustering, fuzzy clustering [9] (also named soft clustering) assigns samples to all the clusters with different probabilities, which can preserve more clustering structure. However, these two kinds of methods rely on the similarity among samples to group the samples, and their performance depends on the quality of the similarity matrix. Thus,

how to learn a discriminative similarity matrix has also become a research focus.

Recently, some subspace theories have provided some new ways to learn the similarity matrix by capturing the subspace structure. Robust principal component analysis (RPCA) [3] is first proposed to decompose the original data into a low-rank matrix, and a sparse noise matrix can enhance the feature by removing the noise. However, RPCA assumes that all the samples are sampled from one subspace, which ignores the multi-subspace structure. Motivated by RPCA, low-rank representation (LRR) [14] is proposed with the assumption that databases with c clusters are sampled from c independent subspaces, and the samples can be linearly represented by the other samples in the same subspaces. However, LRR uses nuclear norm to learn the global structure but ignores the local structure [17].

To preserve the local structure, non-negative low-rank and sparse (NNLRS) [43] representation is proposed to adopt the advantages of nuclear norm and L_1 norm, which can learn the global and local structures in the database. In addition, NNLRS also constrains that the elements in the representation matrix are greater than zero, which makes this matrix can directly show the similarities among the samples. Furthermore, non-negative sparse Laplacian regularized low-rank representation (NSLLRR) [35] introduces a laplacian regularization constructed by the k nearest neighbors and can learn the manifold structure embedded in the database. However, selecting an ideal k is still an open question, which affects the performance of NSLLRR. To address this issue, Fei et al. [8] introduce a distance penalty to LRR to learn the geometric structure in Euclidean space. Based on LRRADP, adaptive weighted non-negative low-rank representation (AWNLRR) [29] is proposed with a sparse weighted matrix to remove the noise, which reduces the bad influence of noise and improves the clustering performance. Hierarchical weighted low-rank representation (HWLRR) [10] improves the LRRADP differently. Based on k nearest neighbors, HWLRR defines the hierarchical weights to learn the hierarchical structures in the data, capturing both global structure and the geometrical structures.

All the low-rank representation methods mentioned above learn the structures of the database without any prior knowledge, but in many clustering tasks, the number of clusters is known. Motivated by this, low-rank with adaptive graph regularization (LRRAGR) [27] and robust spectral ensemble clustering (RSEC) [25] adopt spectral constraints to ensure that the learned non-negative representation matrices contain c connection components. Thus the learned matrices are more suitable for graph-based clustering methods like Ncut [22]. Besides, double low-rank representation with projection distance penalty (DLRRPD) [11] utilizes the number of the clusters to guide the feature projection to gain a better clustering performance.

Although the latest LRR-based methods begin to utilize the spectral constraints to enhance the representation matrices, these methods still need extra clustering algorithms to gain the clustering results. In general, LRR-based clustering methods can be summarized into a two-step framework: 1) learn the representation matrix and construct the similarity matrix; 2) gain the clustering results by the clustering methods. Since these two steps are independent, the learned matrix may not suit the clustering method. To address this problem, we propose a novel LRR model, i.e., One-step Low-Rank Representation (OLRR), to learn the subspace structures and gain the clustering results simultaneously.

The main contributions of this paper are summarized as follows:

(1) Different from existing methods, to learn the clustering structure of databases and gain the clustering result directly, OLRR integrates low-rank representation into clustering by approximating the representation matrix with two same clustering indicator matrices.

(2) Upon the clustering indicator matrix, a probability penalty is introduced to encourage the samples in the same cluster to be similar in the representation space, further improving the discrimination of clustering indicator matrices.

(3) OLRR applies the probability penalty to denoising. Thus the representation and clustering are performed in a clean recovered space for better robustness and performance.

(4) An efficient algorithm is proposed to solve our model, and extensive experimental results show the effectiveness and robustness of our method.

2 NOTATIONS AND RELATED WORK

Notations. The matrix is denoted by the uppercase letter, e.g., X. $X_{:,i}$ and $X_{j,:}$ are the ith column and jth row of the matrix X, respectively. $X_{i,j}$ denotes the element which is on the ith row and jth column of X. $\|X_{i,:}\|_2$ is the L_2 norm of the vector $X_{i,:}$. X^T is the transpose of X. X^{-1} is the inverse of X. $\mathrm{tr}(X)$ is the trace of X. $\|X\|_{2,1}$, $\|X\|_1$, $\|X\|_F$, $\|X\|_*$ and $\|X\|_\infty$ denote $L_{2,1}$ norm, L_1 norm, Frobenius norm, nuclear norm and infinite norm of X, respectively. $\mathbf{1}$ is the column vector in which elements are 1. I is the identity matrix. $\odot$ is the Hadamand product that demotes the element-wise multiplication. $\mathbf{a}$ is a row vector, and $\mathbf{a}_i$ is the ith element of $\mathbf{a}$. $A = \mathrm{Diag}(\mathbf{a})$ is a diagonal matrix in which $A_{i,i} = \mathbf{a}_i$ and $\forall i \neq j, A_{i,j} = 0$. $\mathbf{a} = \mathrm{diag(A)}$ is a row vector defined as $\mathbf{a}_i = A_{i,i}$.

2.1 Low-Rank Representation

For the given noiseless database $X_0 \in R^{d\times n}$ sampled from c independent subspaces, the samples in the same subspace are linearly dependent. Thus this database can be represented by itself with a low-rank representation matrix. Therefore, LRR is proposed to learn this low-rank representation matrix as

$$\min_Z \|Z\|_*, \ \text{s.t.} X_0 = X_0 Z \tag{1}$$

where Z is the representation matrix whose optimal solution is $Z = V_0 V_0^T$, and V_0 can be obtained by skinny SVD of X_0 as $X_0 = U_0 \Sigma_0 V_0^T$. Since there is always noise in the data as $X = X_0 + E_0$, an error matrix is introduced to LRR, and the final LRR model is as

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_1, \ \text{s.t.} X = XZ + E \tag{2}$$

where E is the noise fitting matrix, and $\lambda > 0$ is the balance parameter. LRR learns the low-rank subspace structure but ignores the local structure of data.

2.2 Spectral Clustering

The representation matrix Z is usually handled by the clustering methods to gain the clustering result. Spectral clustering (SC) is

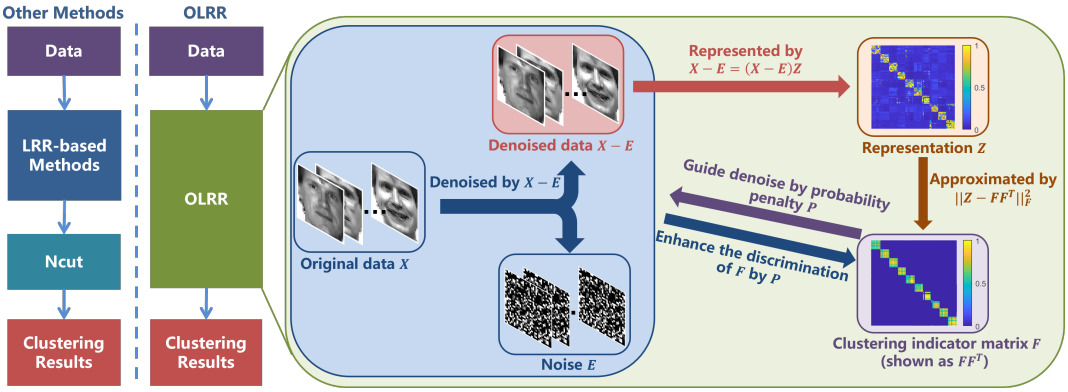


Figure 1: The core difference between OLRR and the exiting method is shown on the left. The flow diagram of our OLRR is given on the right, where we show the relations of denoising, representation, probability penalty and clustering. (P denotes the probability penalty as $\sum_{i,j}\|(X_{:,i}-E_{:,i})-(X_{:,j}-E_{:,j})\|_2^2 F_{i,:}F_{j,:}^T$). Z and FF^T are obtained by Jaffe.

often selected to assign labels for LRR-based methods. For the given affinity matrix A, SC can partition the vertices in the A into c disjoint sets [4]. Y is defined as the cluster indicator matrix, and then the SC can be formulated as

$$\min_{Y\in\{0,1\}} \mathrm{tr}(Y^T L_A Y) \tag{3}$$

where $L_A = D_A - A$ is the Laplacian matrix of A, and D_A is the degree matrix defined as $D_{i,i} = \sum_i A_{i,j}$. However, solving this objective function is an NP-hard question. Thus, some methods this objective function with some additional constraints, and Ratio Cut [21], Normalized Cut (Ncut)[22] and Self-Balanced Min-Cut (SBMC)[5] are three of them. Since SC does not influence the LRR-based methods, the learned representation matrices can be unsuitable for SC [26], leading to worse performance.

3 PROPOSED FORMULATION

OLRR is proposed to combine the clustering and LRR model to improve the performance, and the core novelty and flow diagram of OLRR are shown in Fig. 1.

3.1 Objective Function

A clustering indicator matrix $F \in R^{n\times c}$ is defined that can show the probability of samples belonging to each cluster. Because of the physical meaning of F, we constrain F to satisfy

$$F \geq 0, F_{i,:}\mathbf{1} = 1 \tag{4}$$

where $F_{i,j}$ shows the probability of sample $X_{:,i}$ belonging to the jth cluster. Based on the definition of F, we can easily learn the probability of two different samples $X_{:,i}$ and $X_{:,j}$ belonging to the same cluster as $\sum_{k=1}^{c} F_{i,k}F_{j,k} = F_{i,:}F_{j,:}^T$.

An ideal representation matrix Z can show the affinity among samples, and the samples that have more contribution to representation have higher probabilities from the same cluster[24, 36]. To obtain a better F and Z, we hope that $F_{i,:}F_{j,:}^T \longrightarrow 0$ if $Z_{i,j}$ is small and $F_{i,:}F_{j,:}^T \longrightarrow 1$ if $Z_{i,j}$ is large. We approximate the representation matrix Z with two same fuzzy clustering matrices by $\|Z - FF^T\|_F^2$. Thus, minimizing $\|Z - FF^T\|_F^2$ can ensure that $X_{:,i}$ and $X_{:,j}$ are more likely to be sampled from the same cluster if $X_{:,i}$ has more contribution in representing $X_{:,j}$, i.e., $Z_{i,j}$ is larger. Moreover, to preserve the local structure, a probability penalty is introduced as $\sum_{i,j}\|X_{:,i} - X_{:,j}\|_2^2 F_{i,:}F_{j,:}^T$, where $\|X_{:,i} - X_{:,j}\|_2^2$ denotes the Euclidean distance. Minimizing this term can preserve the local geometric structure by automatically assigning a higher probability to the samples with a smaller distance. And then, the initial problem of OLRR in the noise-free case can be defined as

$$\begin{aligned}\min_{Z,F} \sum_{i,j}\|X_{:,i} - X_{:,j}\|_2^2 F_{i,:}F_{j,:}^T + \lambda_1\|Z\|_* + \frac{\lambda_2}{2}\|Z - \\ FF^T\|_F^2, \text{s.t.} X = XZ, F \geq 0, F_{i,:}\mathbf{1} = 1, \mathrm{diag}(Z) = 0\end{aligned} \tag{5}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are two parameters. By minimizing the clustering constraint, i.e., approximating Z using FF^T, one can avoid the trivial solution as $Z_{i,:} = 0$. $\mathrm{diag}(Z) = 0$ is used to avoid the bad influence of the self-representation.

However, databases always contain various noise and errors in real applications [19, 39], which inaccurate the learned representation and decreases the performance. To tackle this problem, OLRR adopts a robust joint representation. In particular, OLRR first denoises the original X by correcting errors with L_1 norm and then performs representation learning and clustering in a recovered clean space $X - E$, which leads to the final objective function:

$$\begin{aligned}\min_{Z,F,E} \sum_{i,j}\|(X_{:,i} - E_{:,i}) - (X_{:,j} - E_{:,j})\|_2^2 F_{i,:}F_{j,:}^T + \\ \lambda_1\|Z\|_* + \frac{\lambda_2}{2}\|Z - FF^T\|_F^2 + \lambda_3\|E\|_1, \text{s.t.}(X - E) \\ = (X - E)Z, F \geq 0, F_{i,:}\mathbf{1} = 1, \mathrm{diag}(Z) = 0\end{aligned} \tag{6}$$

where $\lambda_3 \geq 0$ is the balance parameter. Minimizing $\sum_{i,j}\|(X_{:,i} - E_{:,i}) - (X_{:,j} - E_{:,j})\|_2^2 F_{i,:}F_{j,:}^T$ and $\|Z - FF^T\|_F^2$ can hold three additional good properties:

- The noise is removed first, improving the robustness of OLRR. Besides, the representing and clustering perform in the clean data space, which improves the clustering performance.
- The processing of denoising is guided by the clustering indicator matrix. With the constraint $(X - E) = (X - E)Z$ and

$\sum_{i,j} \|(X_{:,i} - E_{:,i}) - (X_{:,j} - E_{:,j})\|_2^2 F_{i,:}F_{j,:}^T$, the samples, which have higher possibility from the same cluster, will also be ensured that they are similar in the representation space. In this way, the local structure can also be captured.

- The probability penalty can be reformed as $\|D_{X-E} \odot (FF^T)\|_1$, where $(D_{X-E})_{i,j} = \|(X_{:,i} - E_{:,i}) - (X_{:,j} - E_{:,j})\|_2^2$. Consequently, this term can improve the discrimination of the clustering indicator matrix.

3.2 Clustering via the Clustering Indicator Matrix

The clustering indicator matrix F can be obtained by solving the objective function Eq. (6). Since $F_{i,j}$ denotes the membership of $X_{:,i}$ to the jth cluster, the label of $X_{:,i}$ can be obtained by finding the most related cluster, in other words, finding the index of max elements in $F_{i,:}$. This processing can be formulated as

$$\mathbf{r}_i = k, \text{where } F_{i,k} = \|F_{i,:}\|_\infty \tag{7}$$

where $\mathbf{r}$ is the label vector of X.

Next, the optimization procedures are given.

3.3 Optimization

In this subsection, inexact Augmented Lagrange Multiplier method (ALM) [1] is used to solve the objective function. For convenience, we first rewrite the original formulation Eq. (6) into the following function

$$\begin{aligned}\min_{Z,E,F,J,U,C_1,C_2,C_3} &\sum_{i,j} \|U_{:,i} - U_{:,j}\|_2^2 F_{i,:}F_{j,:}^T + \lambda_1\|J\|_* \\ &+\frac{\lambda_2}{2}\|Z - FF^T\|_F^2 + \lambda_3\|E\|_1 + \frac{\mu}{2}(\|X - U - E + \\ &\frac{C_1}{\mu}\|_F^2 + \|U - UZ + \frac{C_2}{\mu}\|_F^2 + \|Z - J + \frac{C_3}{\mu}\|_F^2)\end{aligned} \tag{8}$$

where J and U are two alternative variables, C_1, C_2 and C_3 are three Lagrange multipliers. μ is the penalty parameter. Then, this problem can be solved by following steps.

Fix others, update E: When other variables are given, E can be obtained by minimizing the following formulation as

$$\min_E \lambda_3\|E\|_1 + \frac{\mu}{2}\|U - X + E + \frac{C_1}{\mu}\|_F^2 \tag{9}$$

and this sub-problem has a closed-form solution [13] as

$$E = \Omega_{\frac{\lambda_3}{\mu}}(X - U + \frac{C_1}{\mu}) \tag{10}$$

where Ω is the shrinkage operator [13].

Fix others, update J: With the other variables fixed, J can be calculated by

$$\min_J \lambda_1\|J\|_* + \frac{\mu}{2}\|Z - J + \frac{C_2}{\mu}\|_F^2 \tag{11}$$

This problem has a closed-form solution as

$$J = \Theta_{\frac{\lambda_1}{\mu}}(Z + \frac{C_2}{\mu}) \tag{12}$$

where Θ is the singular value thresholding (SVT) shrinkage operation [15].

Fix others, update Z: By fixing the other variables, Z can be obtained by solving the following subproblem

$$\begin{aligned}\min_Z \frac{\lambda_2}{2}\|Z - FF^T\|_F^2 + \frac{\mu}{2}(\|U - UZ + \frac{C_2}{\mu}\|_F^2 \\ +\|Z - J + \frac{C_3}{\mu}\|_F^2),\ \text{s.t.diag}(Z) = 0\end{aligned} \tag{13}$$

The solution without constraint $\bar{Z}$ can be obtained as

$$\begin{aligned}\bar{Z} = ((\lambda_2 + \mu)I + \mu U^T U)^{-1}(\lambda_2(FF^T - I) + \\ \mu(U^T(U + \frac{C_2}{\mu}) + J - \frac{C_3}{\mu}))\end{aligned} \tag{14}$$

Then, the final solution of Z can be obtained as

$$Z = \bar{Z} - \text{Diag}(\text{diag}(\bar{Z})) \tag{15}$$

Fix others, update U: When the other variables are fixed, U can be learned by minimizing the following problem

$$\begin{aligned}\min_U \sum_{i,j} \|U_{:,i} - U_{:,j}\|_2^2 F_{i,:}F_{j,:}^T + \frac{\mu}{2}(\|X - U - E + \\ \frac{C_1}{\mu}\|_F^2 + \|U - UZ + \frac{C_2}{\mu}\|_F^2)\end{aligned} \tag{16}$$

For simplify, this problem can be written as

$$\begin{aligned}\min_U 2\text{tr}(UL_FU^T) + \frac{\mu}{2}(\|X - U - E + \frac{C_1}{\mu}\|_F^2 + \\ \|U - UZ + \frac{C_2}{\mu}\|_F^2)\end{aligned} \tag{17}$$

where L_F is the Laplacian matrix of FF^T. Then, U can be obtained as

$$\begin{aligned}U = \mu(X - E + \frac{C_1}{\mu} - \frac{C_2}{\mu}(I - Z^T))(4L_F + \\ \mu I + \mu(I - Z)(I - Z^T))^{-1}\end{aligned} \tag{18}$$

Fix others, update F: As for F, it can be obtained by following problem

$$\begin{aligned}\min_F \text{tr}(D_U^T FF^T) + \lambda_2\|Z - FF^T\|_F^2, \\ \text{s.t.} F \geq 0, F_{i,:}\mathbf{1} = 1\end{aligned} \tag{19}$$

where D_U is the distance matrix defined as $(D_U)_{i,j} = \|U_{:,i} - U_{:,j}\|_2^2$. We use ALM to solve this subproblem, and for convenience, an additional variable G is introduced to separate the problem as

$$\begin{aligned}\min_{F,G,C} \text{tr}(D_U^T FG^T) + \lambda_2\|Z - FG^T\|_F^2 + \\ \frac{\sigma}{2}\|F - G + \frac{C}{\sigma}\|_F^2,\ \text{s.t.} F \geq 0, F_{i,:}\mathbf{1} = 1\end{aligned} \tag{20}$$

where C is the Lagrange multiplier, and σ is the penalty parameter. Then, G can be solved with F fixed as

$$G = (-D_UF + \lambda_2 Z^T F + \mu(F + \frac{C}{\sigma}))L_1^{-1} \tag{21}$$

where $L_1 = \lambda_2 F^T F + \sigma I$. With a given G, a latent $\bar{F}$ without constraint can be calculated as follows.

$$\begin{aligned}\min_F \text{tr}(D_U^T FG^T) + \lambda_2\|Z - FG^T\|_F^2 + \\ \frac{\sigma}{2}\|F - G + \frac{C}{\sigma}\|_F^2\end{aligned} \tag{22}$$

$\bar{F}$ can be directly obtained as

$$\bar{F} = (-D_U G + \lambda_2 ZG + \sigma(G - \frac{C}{\sigma}))L_2^{-1} \tag{23}$$

where $L_2 = \lambda_2 G^T G + \sigma I$, and then F can be obtained by

$$\min_{F \geq 0, F_{i,:}\mathbf{1}=1} \|F - \bar{F}\|_F^2 \tag{24}$$

This problem can be solved row by row, i.e.,

$$F_{i,:} = \max(\xi \mathbf{1} + \bar{F}_{i,:}, 0) \tag{25}$$

where ξ is the Lagrangian multiplier defined as

$$\xi = (1 + \bar{F}_{i,:}\mathbf{1})/(c - 1) \tag{26}$$

The other variables are updated as follows.

$$C = C + \sigma(F - G) \tag{27}$$

$$\sigma = \min(\rho\sigma, \sigma_{\max}) \tag{28}$$

where ρ and $\sigma_{\max}$ are two constants and denote the learning rate and upper bound of σ, respectively. Updating F is summarized as Algorithm 1.

Algorithm 1: Updating F

Input: D_U, Z
Output: F
Initialization: $G = F = C = 0$, $\sigma = 0.01$, $\rho = 1.2$, $\sigma_{\max} = 10^5$.
while *not converged* **do**
  Fix others, update G by Eq. (21);
  Fix others, update F by Eqs. (23) and (25);
  Fix others, update ξ by Eq. (26);
  Update C and σ by Eqs. (27) and (28);
  Convergence check: if $\|F - G\|_\infty < 10^{-5}$, stop.

Update the other variables:

$$C_1 = C_1 + \mu(X - U - E) \tag{29}$$

$$C_2 = C_2 + \mu(U - UZ) \tag{30}$$

$$C_3 = C_3 + \mu(Z - J) \tag{31}$$

$$\mu = \min(\rho\mu, \mu_{\max}) \tag{32}$$

where ρ denotes the learning rate, and $\mu_{\max}$ is the upper bound of μ, respectively. Finally, the algorithm to solve OLRR is summarized as Algorithm 2.

4 ANALYSIS OF OLRR

4.1 Computational Complexity Analysis

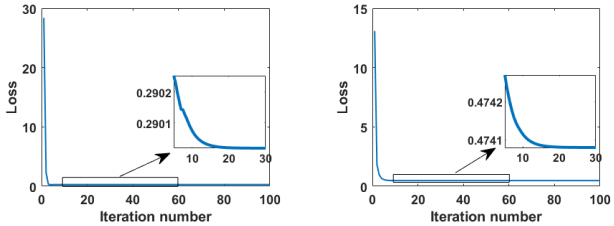
As shown in Algorithm 2, solving OLRR contains five main steps, and we will analyze each step to show the computational complexity of it. E is updated by singular value thresholding, and thus the computational complexity of updating E is $O(n^3)$. Updating J uses eigen-decomposition and its computational complexity is $O(rn^2)$, where r denotes the number of the selected eigenvectors [31]. Inverse operation is the main computational complexity of updating Z, U and F, resulting in that their computational complexities are $O(n^3)$, $O(n^3)$ and $O(2t_1(c^2n + cn^2))$, respectively, where t_1 is the number of Algorithm 1 iteration. Finally, we obtain the whole computational complexity, i.e., $O(t_2(2t_1(c^2n + cn^2) + 3n^3 + rn^2))$, where t_2 is the iteration number of Algorithm 2.

Algorithm 2: Solving OLRR

Input: Database X, the number of clusters c and the tuning parameters λ_1, λ_2, λ_3.
Output: Z, F, E, $\mathbf{r}$.
Initialization: Z constructed by the 10-nearest neighbor graph, $E = 0$, $U = X$, $F = 0$, $J = Z$, $C_1 = 0$, $C_2 = 0$, $C_3 = 0$, $\mu = 0.01$, $\rho = 1.2$, $\mu_{\max} = 10^5$.
while *not converged* **do**
  Fix others, update E by Eq. (10);
  Fix others, update J by Eq. (12);
  Fix others, update Z by Eqs. (14) and (15);
  Fix others, update U by Eq. (18);
  Fix others, update F as Algorithm 1;
  Update C_1, C_2, C_3 and μ by Eqs. (29) to (32);
  Convergence check: if $\max(\|X - U - E\|_\infty, \|U - UZ\|_\infty, \|Z - J\|_\infty) < 10^{-5}$, stop.
Gain the clustering result by Eq. (7).

4.2 Convergence Analysis

As presented before, the objective formulation is solved by the ALM-style method with five blocks, but it is still hard to prove the strict convex for ALM-style methods with more than two blocks [12, 28]. However, optimizing each block will decrease the value of the objective function. Thus, the loss of the objective formulation will also monotonically decrease with fixing others and updating each variable alternately. In this section, we prove the convergence of OLRR by experiments in the following. The objective function loss versus the iteration number on the Auto and Cars database is plotted in Fig. 2. As shown in Fig. 2, the loss of the objective function, i.e., Obj $= (\sum_{i,j} \|(X_{:,i} - E_{:,i}) - (X_{:,j} - E_{:,j})\|_2^2 F_{i,:} F_{j,:}^T + \lambda_1 \|Z\|_* + \frac{\lambda_2}{2}\|Z - FF^T\|_F^2 + \lambda_3 \|E\|_1)/\|X\|_F$, will monotonically decrease and fast converges to a local optimum, which can show the convergence of OLRR.



(a) Auto (b) Cars

Figure 2: Objective function loss versus iteration number of OLRR on the (a) Auto and (b) Cars.

5 EXPERIMENTS AND ANALYSIS

5.1 Experimental Settings

Compared methods. Some most related and state-of-the-art LRR methods, including LRR, NSLLRR [35], AWNLRR [29], LRRAGR

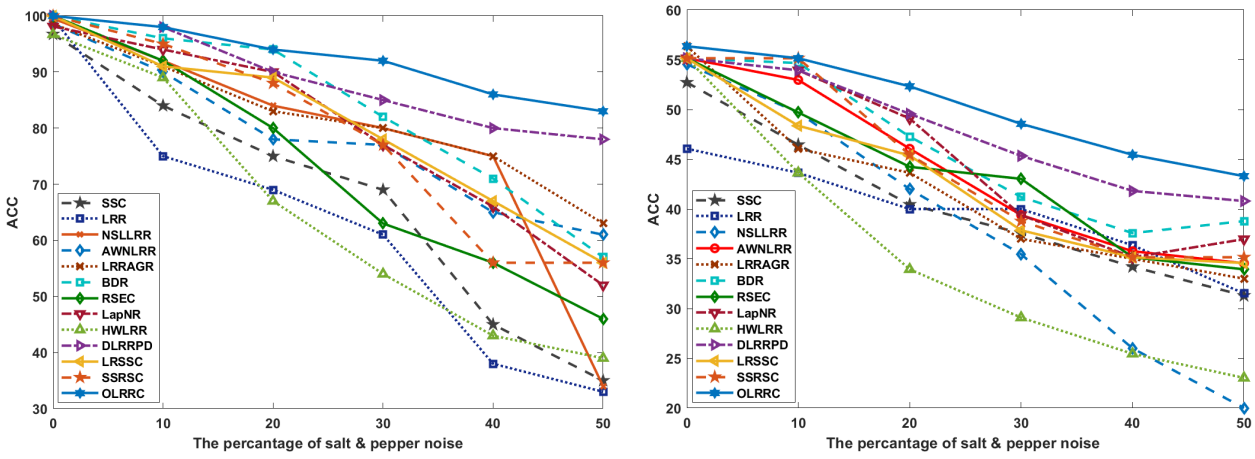
Table 1: Clustering results on real databases

Methods	Metric	Auto	Cars	Control	Glass	Solar	Isolet	Yeast	Dig	USPS	Jaffe	Yale	YaleB
SSC		40.00	61.99	54.11	48.59	40.55	54.29	30.99	14.07	54.52	96.71	52.72	81.23
LRR		40.98	62.76	48.17	53.27	51.39	55.96	30.26	79.13	70.95	99.53	46.06	67.43
NSLLRR		41.95	63.52	65.00	57.48	54.80	59.36	27.09	67.78	65.47	99.53	54.55	38.48
AWNLRR		40.98	66.33	53.17	54.67	55.11	58.40	39.49	79.86	72.10	98.59	55.15	88.06
LRRAGR		39.02	62.76	56.83	55.61	45.51	54.49	30.39	59.32	70.67	98.59	56.36	87.04
BDR		41.95	67.35	55.17	56.54	57.59	61.60	33.09	81.68	67.59	**100**	55.15	70.46
RSEC	ACC	44.39	63.01	54.33	54.67	56.04	62.95	38.01	79.19	68.13	**100**	55.15	88.52
LapNR		41.46	57.14	37.83	53.74	52.63	58.65	39.22	76.02	56.54	98.12	55.15	69.80
HWLRR		37.56	67.86	67.17	51.87	50.77	65.77	40.30	87.37	65.87	96.71	55.15	62.14
DLRRPD		46.83	68.37	76.33	58.41	54.80	67.37	38.54	88.81	69.63	**100**	55.15	63.24
LRSSC		45.14	65.14	72.73	56.34	55.62	65.04	42.56	83.15	71.26	**100**	55.15	82.65
SSRSC		43.41	66.33	61.36	54.67	57.28	58.01	46.16	81.30	70.47	**100**	55.15	86.04
OLRR		**47.04**	**75.77**	**81.67**	**62.15**	**60.37**	**68.53**	**47.44**	**89.20**	**73.80**	**100**	**58.18**	**89.23**
SSC		43.90	62.50	64.33	61.32	47.99	52.15	31.47	14.30	63.92	96.71	52.73	81.48
LRR		45.85	63.01	48.50	60.75	56.66	55.96	31.33	81.75	76.96	99.53	46.06	67.48
NSLLRR		48.78	63.78	66.83	59.81	63.16	59.36	32.14	73.29	58.36	99.53	54.55	38.77
AWNLRR		47.32	68.88	60.83	57.01	57.89	58.40	41.51	81.36	78.63	98.59	55.15	88.69
LRRAGR		41.95	64.54	66.67	60.28	48.92	54.49	31.47	61.77	76.24	98.59	56.36	86.27
BDR		47.32	66.58	51.67	59.81	57.89	61.60	32.08	83.16	71.52	**100**	55.15	70.55
RSEC	Purity	45.85	63.27	60.00	57.01	61.30	62.95	41.51	81.64	74.53	**100**	55.15	88.65
LapNR		47.80	65.05	50.00	63.08	60.37	58.65	49.60	77.46	66.35	98.12	55.15	70.59
HWLRR		41.95	68.88	74.83	62.15	58.51	65.77	48.52	87.37	58.35	96.71	55.15	61.92
DLRRPD		47.80	68.62	76.33	62.62	60.68	67.37	49.60	88.81	75.63	**100**	55.15	64.27
LRSSC		46.52	67.98	75.36	58.62	60.57	65.33	47.53	84.46	77.82	**100**	55.15	81.13
SSRSC		45.78	66.33	61.83	62.55	60.99	58.01	46.93	81.30	77.34	**100**	55.15	86.04
OLRR		**49.53**	**75.77**	**82.41**	**63.08**	**62.54**	**68.53**	**48.32**	**89.20**	**79.18**	**100**	**58.18**	**89.23**
SSC		33.32	62.58	64.49	42.17	31.05	54.29	36.30	10.15	43.97	93.62	34.28	49.58
LRR		34.05	63.17	57.25	40.22	44.82	50.60	30.93	72.85	64.32	99.05	28.25	46.17
NSLLRR		33.48	63.67	62.11	48.70	45.87	51.75	30.35	63.23	56.21	99.03	36.79	13.11
AWNLRR		32.46	66.04	53.47	49.17	46.65	51.53	28.76	76.90	68.14	97.11	36.15	85.13
LRRAGR		34.20	59.25	68.10	51.05	37.84	50.51	32.94	45.25	66.46	97.10	38.68	84.38
BDR		32.76	60.81	67.63	50.96	51.97	53.27	31.12	74.58	64.57	**100**	35.07	38.08
RSEC	Fscore	35.16	58.92	57.61	48.57	47.81	53.43	31.75	72.32	67.34	**100**	35.07	80.81
LapNR		32.36	50.54	54.28	42.63	45.62	52.24	29.39	71.70	50.88	96.32	35.40	53.64
HWLRR		32.22	66.26	70.15	43.60	44.20	54.95	30.56	82.80	67.04	93.95	38.57	42.08
DLRRPD		36.70	66.18	68.85	47.09	44.55	56.15	28.77	83.22	66.53	**100**	38.39	46.35
LRSSC		35.57	67.15	69.53	47.24	46.36	53.56	29.43	78.24	68.25	**100**	38.39	79.24
SSRSC		34.08	59.64	56.34	42.99	47.16	51.94	32.60	73.77	67.34	**100**	38.57	84.25
OLRR		**38.23**	**75.66**	**72.83**	**51.20**	**49.75**	**56.81**	**34.60**	**84.45**	**69.89**	**100**	**39.86**	**86.23**

[27], RSEC [25], BDR [18], LapNR [40], HWLRR [10], DLRRPD [11] and SSRSC [32], are selected as the compared methods to show the effectiveness of our method. Besides, some sparse subpace methods, i.e., SSC and LRSSC [2] are also used as comparison methods. The representation matrices learned by the compared methods are symmetrized by $A = (|Z| + |Z|^T)/2$ and handled by Ncut to gain the final clustering results. Moreover, since these methods are sensitive to the super parameter, we vary each parameter in a wide range to find the best combination. It should be pointed out that the learned representation matrices of compared LRR-based methods are stable with the fixed parameter, but the clustering performance is unstable because the performance of Ncut depends on initializations. Thus, for these compared methods, we report the best results of ten repetitions with the best parameters. As for our method, we ran once with the best parameters because it gained stable clustering results.

Databases. In this section, twelve real databases are used to evaluate the performance of all the methods mentioned above. These databases include seven UCI databases: Auto, Cars, Control, Glass, Solar, Isolet and Yeast, two digit databases: Dig and USPS, and three face databases: Jaffe, Yale and YaleB.

Evaluation metrics. Clustering accuracy (ACC), purity and F-score are used as the evaluation metrics.



(a) Jaffe (b) Yale

Figure 3: Clustering accuracy versus different percentages salt & pepper noise on Jaffe and Yale.

Table 2: Description of the databases

Type	Database	# Class	# Dim	# Sample
UCI	Auto	6	25	205
	Cars	3	8	392
	Control	6	60	600
	Glass	6	9	214
	Solar	6	12	323
	Isolet	2	617	1560
	Yeast	10	1470	1484
Digit	Dig	10	64	1797
	USPS	10	256	1000
Face	Jaffe	10	676	213
	Yale	15	1024	165
	YaleB	38	1024	2414

5.2 Clustering Results and Analysis

From the results in Tab. 1, we can conclude as:

- Our method gains the best performance on these databases. Compared with the second-best method on Cars and Control, our method can achieve about 7.4% and 5.5% improvement in terms of ACC, respectively.
- AWNLRR, LRRAGR, DLRRPD and OLRR often achieve better performance than NSLLRR, proving that preserving the structure by distance penalty is more effective than the k nearest neighbors.
- Compared with other methods, LRRAGR, BDR, RSEC, DLRRPD and OLRR perform better on most databases because they utilize the number of clusters to constrain the representation. OLRR performs best because it combines clustering to boost the performance, proving the effectiveness of combining clustering.
- Compared with DLRRPD that projects the features to a better space to improve the performance, OLRR can achieve better clustering results without feature projection. DLRRPD uses the learned representation to guide the feature projection, but OLRR uses the probability to guide the representation. We can find that the probability is more effective than the representation matrix.

5.3 Clustering Against Noise

OLRR removes the noise and performs the representation in the clean space, improving the robustness of the model. Thus, in this section, we use two databases, i.e., Jaffe and Yale, to show the robustness of OLRR against the salt & pepper noise. All methods are performed on these two databases with different percentage noise, and the percentages of the noise are $[0, 10, 20, 30, 40, 50]$. ACC is used to evaluate the performance, and the final performance is shown in Fig. 3. As shown in Fig. 3, all the methods perform worse as the level of the occupation increase. Besides, our OLRR gains the best performance with all level noise, which shows OLRR is more robust than comparison methods against salt & pepper noise.

5.4 Ablation Experiments

Some ablation experiments are conducted to evaluate the effectiveness of each improvement. We use I_1, I_2 and I_3 to denote the three main improvements of our model, where I_1 denotes denoising, I_2 is the $\|Z - FF^T\|_F^2$ and I_3 is the probability penalty. The effectiveness of each improvement is evaluated by performing on databases without this improvement.

As shown in Tab. 3, we can find that the performance of OLRR-I_i is worse than OLRR, which can show the effectiveness of each improvement. Specifically, OLRR-I_2 gains the worst performance in most cases because the learned clustering indicator matrix F can only capture the distance relationship without the constraint $\|Z - FF^T\|_F^2$. Thus, OLRR-I_2 has significant performance degradations on Jaffe10% and Jaffe20% because the distance relationship is destroyed. Compared with the performance on Jaffe, OLRR-I_1 has a larger

Table 3: Effectiveness of each improvement measured by ACC, and OLRR$-I_i$ denotes the performance without I_i.

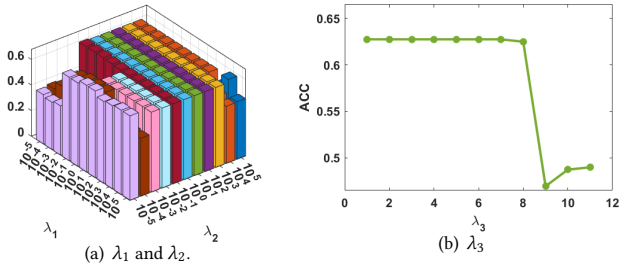
Dataset	OLRR$-I_1$	OLRR$-I_2$	OLRR$-I_3$	OLRR
Auto	44.38	33.17	42.44	47.04
Cars	71.17	62.50	71.19	75.77
Jaffe	97.18	96.71	87.32	100
Jaffe10%	86.85	71.83	85.92	98.12
Jaffe20%	84.98	64.79	78.87	93.90

*Jaffe10% and Jaffe20% denote the Jaffe with 10% and 20% salt & pepper noise respectively.

performance degradation on Jaffe10% and Jaffe20%, which can show that I_1, i.e., denoising, can enhance the robustness of our model.

5.5 Parameter Sensitivity and Selection

From the objective function Eq. (6), we can find that there are three parameters, i.e., λ_1, λ_2 and λ_3. These parameters balance the effect of different terms, where λ_1, λ_2 and λ_3 balance the low-rank constraint, matrix approximation and the noise fitting, respectively. Since OLRR with different combination of three parameters achieves different performance, we test the sensitivity of each parameter to select the best combination of parameter. For convenience, OLRR is performed on the Cars database with the different combination of three parameters, and each parameter is varied in the range $[10^{-5}, 10^{-4}, ..., 10^3, 10^4, 10^5]$. First, λ_1 is fixed as $\lambda_3 = 10^0$, and λ_1 and λ_2 are tuned. Thus, the clustering performance on different combinations of λ_1 and λ_2 is shown in Fig. 4(a). From Fig. 4(a), it can be found that OLRR can deliver good results $\lambda_2 \geq 10^{-2}$ and OLRR is robust on λ_1. Then, we fix $\lambda_1 = 10^{-2}$ and $\lambda_2 = 10^{-1}$, and OLRR performs with different λ_3 to show the influence of λ_3. As shown in Fig. 4(b), OLRR can deliver good results with $\lambda_3 \leq 10^3$. Moreover, one can find that λ_2 and λ_3 are insensitive in some extent, thus one can select the two parameters λ_2 and λ_3 from the range of $[10^{-2}, 10^3]$ and $[10^{-3}, 10^2]$ to achieve a good performance.

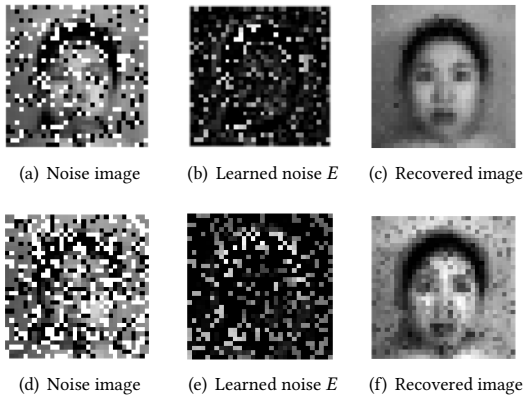


(a) λ_1 and λ_2. (b) λ_3

Figure 4: ACC of our method w.r.t (a) λ_1 and λ_2 with $\lambda_3 = 1$ and (b) λ_3 with $\lambda_1 = 10^{-2}$ and $\lambda_2 = 10^{-1}$ on the Cars.

5.6 Effectiveness of Denoise

In our model, the constraint, i.e., $(X - Z) = (X - Z)Z$, removes the noise of data. Thus, our model is robust against the salt & pepper noise. In this subsection, we give some examples to show the effectiveness of denoise. The noised images, learned noise, and recovered images are obtained by X, E and $X - E$ and are visualized as Fig. 5, where Jaffe with 30% and 50% salt & pepper noise are used as examples. It is obvious that our method can remove the noise and effectively recover the noise image.



(a) Noise image (b) Learned noise E (c) Recovered image

(d) Noise image (e) Learned noise E (f) Recovered image

Figure 5: Noise image, learned noise and recovered image obtained from Jaffe with 30% and 50% salt & pepper noise.

6 CONCLUSION

This paper proposes a novel LRR based model, i.e., One-step Low-Rank Representation (OLRR), for clustering. It approximates the representation matrix with two same clustering indicator matrices, thus gaining the clustering results directly without additional clustering methods. Upon such a clustering indicator matrix, a probability penalty is introduced to capture the local structure of data, resulting in a more discriminative clustering indicator matrix. Finally, by removing the noise occupied in the original data, representation and clustering are then performed in a clean space, thus facilitating a more robust model with better performance. The effectiveness of our OLRR has been evaluated on several benchmark databases for data clustering. The clustering of the data with salt & pepper noise can also show the robustness of our method.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China (No.2021ZD0112100), National NSF of China (No.U1936212, No.62120106009, No.62106123), and the China Postdoctoral Science Foundation (No.2021T140377).

REFERENCES

[1] Stephen P. Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.* 3, 1 (2011), 1–122.
[2] Maria Brbic and Ivica Kopriva. 2020. ℓ_0-Motivated Low-Rank Sparse Subspace Clustering. *IEEE Trans. Cybern.* 50, 4 (2020), 1711–1725.
[3] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. 2011. Robust principal component analysis? *J. ACM* 58, 3 (2011), 1–37.
[4] Xiaojun Chen, Weijun Hong, Feiping Nie, Joshua Zhexue Huang, and Li Shen. 2020. Enhanced Balanced Min Cut. *Int. J. Comput. Vis.* 128, 7 (2020), 1982–1995.
[5] Xiaojun Chen, Joshua Zhexue Huang, Feiping Nie, Renjie Chen, and Qingyao Wu. 2017. A Self-Balanced Min-Cut Algorithm for Image Clustering. In *Int. Conf. Comput. Vis.* 2080–2088.

[6] Yongyong Chen, Shuqin Wang, Chong Peng, Zhongyun Hua, and Yicong Zhou. 2021. Generalized Nonconvex Low-Rank Tensor Approximation for Multi-View Subspace Clustering. *IEEE Trans. Image Process.* 30 (2021), 4022–4035.

[7] Yewang Chen, Lida Zhou, Nizar Bouguila, Cheng Wang, Yi Chen, and Jixiang Du. 2021. BLOCK-DBSCAN: Fast clustering for large scale data. *Pattern Recognit.* 109 (2021), 107624.

[8] Lunke Fei, Yong Xu, Xiaozhao Fang, and Jian Yang. 2017. Low rank representation with adaptive distance penalty for semi-supervised subspace classification. *Pattern Recognit.* 67 (2017), 252–262.

[9] Qiying Feng, Long Chen, C. L. Philip Chen, and Li Guo. 2020. Deep Fuzzy Clustering - A Representation Learning Approach. *IEEE Trans. Fuzzy Syst.* 28, 7 (2020), 1420–1433.

[10] Zhiqiang Fu, Yao Zhao, Dongxia Chang, and Yiming Wang. 2021. A hierarchical weighted low-rank representation for image clustering and classification. *Pattern Recognit.* 112 (2021), 107736.

[11] Zhiqiang Fu, Yao Zhao, Dongxia Chang, Xingxing Zhang, and Yiming Wang. 2021. Double Low-Rank Representation With Projection Distance Penalty for Clustering. In *IEEE Conf. Comput. Vis. Pattern Recog.* 5320–5329.

[12] Zhouchen Lin, Risheng Liu, and Huan Li. 2015. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. *Mach. Learn.* 99, 2 (2015), 287–325.

[13] Zhouchen Lin, Risheng Liu, and Zhixun Su. 2011. Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation. In *Adv. Neural Inform. Process. Syst.* 612–620.

[14] Guangcan Liu, Zhouchen Lin, and Yong Yu. 2010. Robust Subspace Segmentation by Low-Rank Representation. In *Proc. Int. Conf. Mach. Learn.* 663–670.

[15] Guangcan Liu and Shuicheng Yan. 2011. Latent Low-Rank Representation for subspace segmentation and feature extraction. In *Int. Conf. Comput. Vis.* 1615–1622.

[16] Hongfu Liu, Ming Shao, Sheng Li, and Yun Fu. 2016. Infinite Ensemble for Image Clustering. In *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.* 1745–1754.

[17] Zhonghua Liu, Weihua Ou, Wenpeng Lu, and Lin Wang. 2019. Discriminative feature extraction based on sparse and low-rank representation. *Neurocomputing* 362 (2019), 129–138.

[18] Canyi Lu, Jiashi Feng, Zhouchen Lin, Tao Mei, and Shuicheng Yan. 2019. Subspace Clustering by Block Diagonal Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 2 (2019), 487–501.

[19] Feiping Nie, Danyang Wu, Rong Wang, and Xuelong Li. 2020. Self-Weighted Clustering With Adaptive Neighbors. *IEEE Trans. Neural Networks Learn. Syst.* 31, 9 (2020), 3428–3441.

[20] Yong Peng, Xin Zhu, Feiping Nie, Wanzeng Kong, and Yuan Ge. 2021. Fuzzy graph clustering. *Inf. Sci.* 571 (2021), 38–49.

[21] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 5500 (2000), 2323–2326.

[22] Jianbo Shi and Jitendra Malik. 2000. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 8 (2000), 888–905.

[23] Kun Song, Xiwen Yao, Feiping Nie, Xuelong Li, and Mingliang Xu. 2021. Weighted bilateral K-means algorithm for fast co-clustering and fast spectral clustering. *Pattern Recognit.* 109 (2021), 107560.

[24] Yu Song and Yiquan Wu. 2018. Subspace clustering based on latent low rank representation with Frobenius norm minimization. *Neurocomputing* 275 (2018), 2479–2489.

[25] Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. 2019. Robust Spectral Ensemble Clustering via Rank Minimization. *ACM Trans. Knowl. Discov. Data* 13, 1 (2019), 1–25.

[26] Yang Wang, Lin Wu, Xuemin Lin, and Junbin Gao. 2018. Multiview Spectral Clustering via Structured Low-Rank Matrix Factorization. *IEEE Trans. Neural Networks Learn. Syst.* 29, 10 (2018), 4833–4843.

[27] Jie Wen, Xiaozhao Fang, Yong Xu, Chunwei Tian, and Lunke Fei. 2018. Low-rank representation with adaptive graph regularization. *Neural Networks* 108 (2018), 83–96.

[28] Jie Wen, Na Han, Xiaozhao Fang, Lunke Fei, Ke Yan, and Shanhua Zhan. 2019. Low-Rank Preserving Projection Via Graph Regularized Reconstruction. *IEEE Trans. Cybern.* 49, 4 (2019), 1279–1291.

[29] Jie Wen, Bob Zhang, Yong Xu, Jian Yang, and Na Han. 2018. Adaptive weighted nonnegative low-rank representation. *Pattern Recognit.* 81 (2018), 326–340.

[30] Jie Wen, Zheng Zhang, Zhao Zhang, Lei Zhu, Lunke Fei, Bob Zhang, and Yong Xu. 2021. Unified Tensor Framework for Incomplete Multi-view Clustering and Missing-view Inferring. In *Proc. AAAI Conf. Artif. Intell.* 10273–10281.

[31] Jie Wen, Zuofeng Zhong, Zheng Zhang, Lunke Fei, Zhihui Lai, and Runze Chen. 2020. Adaptive Locality Preserving Regression. *IEEE Trans. Circuit Syst. Video Technol.* 30, 1 (2020), 75–88.

[32] Jun Xu, Mengyang Yu, Ling Shao, Wangmeng Zuo, Deyu Meng, Lei Zhang, and David Zhang. 2021. Scaled Simplex Representation for Subspace Clustering. *IEEE Trans. Cybern.* 51, 3 (2021), 1493–1505.

[33] Meixiang Xu, Zhenfeng Zhu, Xingxing Zhang, Yao Zhao, and Xuelong Li. 2020. Canonical Correlation Analysis With $L_{2,1}$-Norm for Multiview Data Representation. *IEEE Trans. Cybern.* 50, 11 (2020), 4772–4782.

[34] Rui Xu and Donald C. Wunsch II. 2005. Survey of clustering algorithms. *IEEE Trans. Neural Networks* 16, 3 (2005), 645–678.

[35] Ming Yin, Junbin Gao, and Zhouchen Lin. 2016. Laplacian Regularized Low-Rank Representation and Its Applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 3 (2016), 504–517.

[36] Ming Yin, Junbin Gao, Zhouchen Lin, Qinfeng Shi, and Yi Guo. 2015. Dual Graph Regularized Latent Low-Rank Representation for Subspace Clustering. *IEEE Trans. Image Process.* 24, 12 (2015), 4918–4933.

[37] Xingxing Zhang, Shupeng Gui, Zhenfeng Zhu, Yao Zhao, and Ji Liu. 2020. Hierarchical Prototype Learning for Zero-Shot Recognition. *IEEE Trans. Multim.* 22, 7 (2020), 1692–1703.

[38] Xingxing Zhang, Zhenfeng Zhu, Yao Zhao, and Deqiang Kong. 2018. Self-Supervised Deep Low-Rank Assignment Model for Prototype Selection. In *Proc. Int. Joint Conf. Artif. Intell.* 3141–3147.

[39] Zhao Zhang, Jiahuan Ren, Sheng Li, Richang Hong, Zhengjun Zha, and Meng Wang. 2019. Robust Subspace Discovery by Block-diagonal Adaptive Locality-constrained Representation. In *Proc. ACM Int. Conf. Multimedia.* 1569–1577.

[40] Yin-Ping Zhao, Long Chen, and C. L. Philip Chen. 2021. Laplacian Regularized Nonnegative Representation for Clustering and Dimensionality Reduction. *IEEE Trans. Circuit Syst. Video Technol.* 31, 1 (2021), 1–14.

[41] Shuai Zheng, Zhenfeng Zhu, Xingxing Zhang, Zhizhe Liu, Jian Cheng, and Yao Zhao. 2020. Distribution-Induced Bidirectional Generative Adversarial Network for Graph Representation Learning. In *IEEE Conf. Comput. Vis. Pattern Recog.* 7222–7231.

[42] Xiaofeng Zhu, Shichao Zhang, Yonghua Zhu, Wei Zheng, and Yang Yang. 2020. Self-weighted Multi-view Fuzzy Clustering. *ACM Trans. Knowl. Discov. Data* 14, 4 (2020), 48:1–48:17.

[43] Liansheng Zhuang, Haoyuan Gao, Zhouchen Lin, Yi Ma, Xin Zhang, and Nenghai Yu. 2012. Non-negative low rank and sparse graph for semi-supervised learning. In *IEEE Conf. Comput. Vis. Pattern Recog.* 2328–2335.