Learning an Optimal Bipartite Graph for Subspace Clustering via Constrained Laplacian Rank

Feiping Nie, *Senior Member, IEEE*, Wei Chang, Rong Wang, and Xuelong Li, *Fellow, IEEE*

***Abstract*—In this article, we focus on utilizing the idea of co-clustering algorithms to address the subspace clustering problem. In recent years, co-clustering methods have been developed greatly with many important applications, such as document clustering and gene expression analysis. Different from the traditional graph-based methods, co-clustering can utilize the bipartite graph to extract the duality relationship between samples and features. It means that the bipartite graph can obtain more information than other traditional graph methods. Therefore, we proposed a novel method to handle the subspace clustering problem by combining dictionary learning with a bipartite graph under the constraint of the (normalized) Laplacian rank. Besides, to avoid the effect of redundant information hiding in the data, the original data matrix is not used as the static dictionary in our model. By updating the dictionary matrix under the sparse constraint, we can obtain a better coefficient matrix to construct the bipartite graph. Based on Theorem 2 and Lemma 1, we further speed up our algorithm. Experimental results on both synthetic and benchmark datasets demonstrate the superior effectiveness and stability of our model.**

***Index Terms*—Co-clustering structure, Laplacian rank constraint, optimal bipartite graph, sparse coefficient, subspace clustering.**

I. Introduction

With the development and popularization of computer technology, the amount of data accumulated is increasing exponentially. Traditional data processing and understanding approaches cannot work very well on this kind of dataset. There are two main reasons for this situation. On the one hand, large-scale data lead to the high processing complexity of data. On the other hand, the collected data generally have the property of the high dimension like the image dataset [1] and movie rating [2], which makes the traditional distance measurements no longer applicable. To solve these problems, finding a concise representation by utilizing the internal structure of data is very crucial for analyzing the essence of data with minimal complexity.

There is a well-known hypothesis that the high-dimensional data can be modeled as samples drawn from the union of multiple low-dimensional linear subspaces with each subspace corresponding to one class or category. Subspace clustering plays an important role in unsupervised learning with numerous applications, such as pattern recognition [3], [4]; image representation and compression [5], [6]; multiagent systems [7]; and computer vision that includes image [1], hand-written digits [8], and video [9] segmentation. In the last decade, subspace clustering has long been a fundamental topic of research and attention while providing new ideas for the development of other technologies, such as dimensionality reduction [10], [11] and multiview clustering [12], [13]. For high-dimensional data clustering problems, the subspace-based methods perform very well in the real scenario.

Prior Work on Subspace Clustering: During the past few decades, there are numerous applications in computing vision and machine learning, etc. Subspace clustering has been extensively studied and a lot of techniques are proposed to address this problem. According to the mechanisms of these subspace clustering techniques, the existing works can be roughly divided into four main categories: 1) algebraic; 2) factorization; 3) statistical learning; and 4) spectral clustering. Algebraic-based method, such as the generalized principal analysis (GPCA) [14], [15], is proposed to utilize the gradient of a multinomial at one point to construct a subspace consisting of that point. Thus, it is equivalent to fitting and differentiating multinomials. Factorization-based approaches, such as [16] and [17], need to find a representation for the given data matrix. This representation is made up of the multiplication of two matrices so that the clusters of data points can be revealed by the support pattern of one factor in the representation. These methods are designed to modify the popular factor analysis algorithms that can produce such factorizations. However, the mentioned approaches cannot have good performance when the data are corrupted. Methods based on statistical learning, such as mixtures of probabilistic principal component analysis (PCA) [18] and multistage learning [16], [19], assume that mixed data are produced by a mixture of Gaussian distributions that are corresponding to each subspace in data, respectively. Generally speaking, these statistics-based methods can be regarded as a mixture of Gaussian approaches.

Manuscript received 12 November 2020; revised 25 March 2021 and 27 June 2021; accepted 2 September 2021. Date of publication 12 October 2021; date of current version 13 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61936014 and Grant 61772427, and in part by the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University under Grant CX2021088. This article was recommended by Associate Editor Y. Tan. *(Corresponding author: Feiping Nie.)*

Feiping Nie, Wei Chang, and Xuelong Li are with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: feipingnie@gmail.com; hsomewei@gmail.com; li@nwpu.edu.cn).

Rong Wang is with the School of Cybersecurity and the School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: wangrong07@tsinghua.org.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3113520.

Digital Object Identifier 10.1109/TCYB.2021.3113520

Recently, the graph-based technique [20], [21] has been applied in the subspace clustering and shows a good performance in handling this problem, such as the sparse learning sparse subspace clustering (SSC) [22] and low-rank representation (LRR) [23]. This kind of method contains two steps that are dictionary representation and spectral clustering. Some unified frameworks are also proposed to incorporate these two processes into a single model and perform better on real-world datasets, such as block-diagonal representation (BDR) [24] and LOSBG [25]. However, for these graph-based methods, the block-diagonal structure for the graph may not be obtained easily by the dictionary representation. There are two main reasons leading to this defect. First, in the process of dictionary representation, the entire original dataset is usually regarded as the fixed dictionary to obtain the coefficient matrix. For the real-world datasets, due to the redundant information hiding in the data, the coefficient matrix learned by this kind of dictionary may not construct an optimal graph with a clear structure. Second, though the learned graph can be constrained by some block-diagonal induced regularizers to capture the ideal structure, it cannot be optimized by algorithms very well due to the nonconvex regularizers, such as the BDR model. Therefore, it is necessary to find an appropriate way to handle these two problems.

A. Paper Contributions

In this article, we proposed a novel representation model that can learn an optimal bipartite graph with the constraint of Laplacian rank to solve the subspace clustering problem. The main contributions of this article are listed as follows.

1) Different from the traditional methods, such as SSC and LRR, we replace the entire data matrix with partial samples from the dataset as the dictionary initialization. Through updating the dictionary matrix, we tactically avoid the trivial solutions for some datasets with the intersecting clusters, and a better coefficient matrix can be learned to construct the graph.
2) Since the updated dictionary is not a square matrix, combining with the idea of co-clustering, we introduce the bipartite graph in our model. So, we can utilize the nonsquare coefficient matrix to build the graph directly. Besides, the bipartite graph can make full use of the duality between samples and features to help our model obtain more latent information in data. Under the constraint of Laplacian rank, the learned bipartite graph can capture the subspace structures more effectively.
3) To solve the proposed model under the rank constraint, a simple yet effective algorithm is designed to transform the rank constraint into an efficient optimization problem. Hence, we can solve the proposed model efficiently and obtain a better block-diagonal structure for the learned bipartite graph. In addition, we combine the constraint condition with the structure of the bipartite graph to further speed up the proposed algorithm.

For the previous work SOBG [20], it needs an initial graph as the input. Hence, SOBG cannot handle the clustering problem directly. However, the proposed model unifies the graph learning method and dictionary representation into a framework, which means that our model is more efficient in handling the subspace clustering problem. Several experiments are conducted to verify the effectiveness and robustness of the proposed model. The results on benchmark datasets show that our algorithm has equivalent performance or is even better than other related methods.

Notations: In the entire article, we define all the matrices by capital letters like M, vectors by the bold lowercase letters or uppercases with a subscript, for example, $\boldsymbol{x}$, $\boldsymbol{x}_i$ or X_i, and the scalars are defined as the lowercase letters like m, m_j. For matrix S, the ith column vector of S is denoted by S_i or $\boldsymbol{s}_i$ and the ijth element is defined as s_{ij}. Besides, the ith row vector of S is denoted by S^i or $\boldsymbol{s}^i$. The trace of matrix S is defined as $Tr(S)$. Some definitions of the norm are given, including the l_1 norm $\|S\|_1 = \sum_{ij} |s_{ij}|$, vector $\boldsymbol{x}$'s l_2 norm $\|\boldsymbol{x}\|_2$, matrix S's Frobenius norm $\|S\|_F$, and nuclear norm $\|S\|_*$. In particular, notation N is defined as a scalar to represent the size of the graph.

II. Related Work

A. Model Based on the Dictionary Representation

There are two typical algorithms based on representations SSC and LRR. Hence, in this section, we will give a concrete description of these two algorithms. SSC and LRR are both representation-based spectral clustering algorithms. For SSC, it consists of two steps' algorithms. The first step is based on the sparse dictionary representation aiming to obtain the adjacency matrix for the graph. Usually, the representation can be obtained by l_1-minimization [26], [27]. Elhamifar and Vidal [22] proposed the SSC model to find a sparse Z by l_1 norm as

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

Here, X is the given data matrix, and SSC selects the data matrix X as the dictionary. In theory, it can be proved the feasible solution Z obtained by SSC has the block-diagonal property. Then, the next step is exploiting adjacency matrix Z to construct the graph G, where $G = (|Z| + |Z^T|)/2$, and using spectral clustering on the graph to obtain the classification results such as the normalized cuts (NCuts) [28].

Another important spectral-type technique is LRR which is similar to SSC. The difference between them is that LRR employs the low-rank representation solved by minimization of the nuclear norm [29]–[31] to obtain the adjacent matrix Z

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

LRR has a unique closed-form solution $Z = VV^T$, in which V is the right singular matrix drawn from the SVD of $X = USV^T$. Besides, the obtained solution Z from LRR satisfies the block-diagonal property when subspaces are independent. Finally, LRR also utilizes the formula $G = (|Z| + |Z^T|)/2$ to construct the graph, and spectral clustering is used for subspace segmentation as same as SSC.

B. Spectral Clustering on Bipartite Graph Revisited

Spectral clustering is a traditional approach for the subspace clustering problem. In the general case, we utilize the distance norm to construct a weighted graph that is used for spectral clustering. However, for the coefficient matrix obtained by the dictionary representation method, it is undoubtedly a better idea to obtain the clustering result by making the full correlation between features and samples. This kind of idea is called co-clustering and Dhillon proposed a very effective model called BSGP [32] for it. In this section, we will revisit the procedure of the method BSGP.

For the given coefficient matrix $Z \in R^{m\times n}$, we need to utilize it to construct the bipartite graph S. The graph can be viewed as an undirected weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{S}\}$ with $N = m + n$ nodes. Here, $\mathcal{V}$ is defined as the node set and the graph S is denoted as follows:

$$S = \begin{bmatrix} 0 & Z \\ Z^T & 0 \end{bmatrix}. \tag{3}$$

Hence, we obtain the bipartite graph $\mathcal{G}$. Assume that graph $\mathcal{G}$ is partitioned into k components. Based on bipartite spectral graph partitioning [32], the NCut on graph $\mathcal{G}$ is equivalent to the trace norm minimization problem as follows:

$$\min_{F^TF=I} Tr\left(F^T\tilde{L}F\right) \tag{4}$$

where $\tilde{L} = I - D^{-(1/2)}SD^{-(1/2)}$ is the normalized Laplacian matrix. D is the diagonal degree matrix denoted as $d_{ii} = \sum_i s_{ij}$.

The matrices F and D can be rewritten as the following form:

$$F = \begin{bmatrix} U \\ V \end{bmatrix}, \ D = \begin{bmatrix} D_u & \\ & D_v \end{bmatrix} \tag{5}$$

where $U \in R^{m\times k}$, $V \in R^{n\times k}$, $D_u \in R^{m\times m}$, and $D_v \in R^{n\times n}$.

Due to the special structure of the graph S defined in (3), problem (4) can be converted into the following problem:

$$\max_{U^TU+V^TV=I} Tr\left(U^TD_u^{-\frac{1}{2}}ZD_v^{-\frac{1}{2}}V\right). \tag{6}$$

To solve problem (6), Nie *et al.* [20] proposed a solvable algorithm with Lemma 1 after removing the discrete constraint for U and V.

Lemma 1: Suppose $M \in R^{n_1\times n_2}$, $X \in R^{n_1\times k}$, and $Y \in R^{n_2\times k}$. The optimal solutions to the problem

$$\max_{X^TX+Y^TY=I} Tr\left(X^TMY\right) \tag{7}$$

are $X = (\sqrt{2}/2)U_1$ and $Y = (\sqrt{2}/2)V_1$. Here, U_1 and V_1 are corresponding to the largest k left and right singular vectors for M.

Based on Lemma 1, the optimal solutions U and V to problem (6) are the leading k left and right singular vectors of $\tilde{Z} = D_u^{-(1/2)}ZD_v^{-(1/2)}$, respectively. Since the solutions U and V may not satisfy the discrete constraint, k-means can be utilized on the rows of F defined in (5) to obtain the final segmentation results.

III. Dictionary Representation With Bipartite Graph

In this section, based on the above introduction, we will propose a novel dictionary representation-based model combining with an optimal bipartite graph. Given a dataset, we denote it as a data matrix $X \in R^{d\times n}$, of which each column $\boldsymbol{x}_i$, $(i = 1\cdots n)$ is a data point. Based on the dictionary representation, each point can be represented by the linear combination of basis in a given dictionary $A = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m] \in R^{d\times m}$

$$X = AZ \tag{8}$$

where X is the dataset, $Z = [z_1, z_2, \ldots, z_n] \in R^{m\times n}$ is the coefficient matrix, and each column z_i is linear representation for the corresponding data point $\boldsymbol{x}_i$ by dictionary matrix A. For the selection of dictionary A, some classical methods, such as LRR [23] and SSC [22], utilize the complete data matrix as the dictionary and have a good performance on the subspace clustering problem. However, this kind of dictionary is often large and redundant, so multiple trivial solutions can be obtained for problem (8).

Based on the subspace clustering hypothesis [22], we know that the subspaces corresponding to the clusters in data are low-rank and independent of each other. Hence, to avoid the redundant problem for dictionary, we intend to select some samples randomly from the dataset to build the dictionary matrix A instead of the entire dataset, which means the number of columns in dictionary m is much less than the number of columns in the dataset n. In this way, the problem of dictionary redundancy can be solved effectively.

According to (8), when the dictionary matrix A is fixed, the purpose of solving the subspace clustering problem is to obtain the optimal coefficient matrix Z. However, we do not know how to choose the basis of the dictionary from the dataset to avoid the appearance of trivial solutions. So, in this situation, we need to update the dictionary matrix A to obtain a better dictionary. Therefore, from what has been discussed above, the objective function can be described as follows:

$$\begin{aligned} &\min_{A,Z} \ \|X - AZ\|_F^2 \\ &\text{s.t.} \ \ Z^T\mathbf{1} = \mathbf{1}, \ Z \geq 0. \end{aligned} \tag{9}$$

Here, $\mathbf{1}$ is a column vector of which the elements are all ones. In the general subspace clustering problem, not all subspace passes through the origin which can be seen in [33]. That means some subspaces are affine subspaces. Based on the properties of affine subspace, the first constraint condition of the coefficient matrix Z is given in problem (9).

According to the classical algorithms SSC [22] and LRR [23], these methods transfer the subspace clustering problem into a two-step algorithm, including building the affinity matrix and spectral clustering. The coefficient matrix Z can be obtained by problem (9). In constructing the graph, LRR and SSC utilize the formula $G = (|Z| + |Z^T|)/2$ to obtain the graph matrix G, which can be easily calculated and obtained. However, this kind of method does not fully consider the relationship between features and samples. From the literature [20], we know the co-clustering methods can make use of the duality between features and samples such that the

co-clustering structure of sample and feature clusters can be extracted. So, in this article, we propose to construct a bipartite graph through the coefficient matrix Z

$$G = \begin{bmatrix} 0 & Z \\ Z^T & 0 \end{bmatrix}. \tag{10}$$

Here, $G \in R^{N\times N}$, $(N = m + n)$ is the bipartite graph and $Z \in R^{m\times n}$ is the coefficient matrix. From the constraint condition of problem (9), we have $Z \geq 0$. Therefore, in (10), we do not need another step to solve the absolute value of matrix Z.

We can see that the optimal bipartite graph does not have a very clear clustering structure. Therefore, we want the bipartite graph G can obtain the form of the block-diagonal structure by learning an optimal coefficient matrix Z when the sample points are ordered. If graph G has exact k connected components, we can obtain the subspace clustering result directly based on the structural graph G.

Hence, to let the graph G have k connected components, the optimal problem (9) can be rewritten as

$$\begin{aligned} \min_{A,Z,G\in\Omega} \ & \|X - AZ\|_F^2 \\ \text{s.t.} \ & Z^T\mathbf{1} = \mathbf{1}, \ Z \geq 0. \end{aligned} \tag{11}$$

Here, Ω represents the set of the graph matrix G, which has exact k connected components.

According to the constraint condition in problem (11), the similarity graph G is non-negative, so the Laplacian matrix $L_G = D_G - G$ associated with G has an important property as follows [34], [35].

Theorem 1: The multiplicity k of the eigenvalue 0 of the Laplacian matrix L_G is equal to the number of connected components in the graph associated with G.

Based on Theorem 1, we know that if the rank of Laplacian matrix L_G associated with G is $N - k$, the bipartite graph G will have k connected components. Hence, the optimal problem (11) can be rewritten as

$$\begin{aligned} \min_{A,Z} \ & \|X - AZ\|_F^2 \\ \text{s.t.} \ & Z^T\mathbf{1} = \mathbf{1}, \ Z \geq 0, \ \text{rank}(L_G) = N - k. \end{aligned} \tag{12}$$

Assume that $\sigma_i(L_G)$ is the ith smallest eigenvalue of L_G. Because the Laplacian matrix L_G is positive semidefinite, $\sigma_i(L_G) \geq 0$. Based on the above conclusion, problem (12) can be equivalent to the following problem:

$$\begin{aligned} \min_{A,Z} \ & \|X - AZ\|_F^2 + \lambda \sum_{i=1}^{k} \sigma_i(L_G) \\ \text{s.t.} \ & Z^T\mathbf{1} = \mathbf{1}, \ Z \geq 0. \end{aligned} \tag{13}$$

Due to $\sigma_i(L_G) \geq 0$ for every i, when λ is very large, the optimal problem (13) will make the last term $\sum_{i=1}^{k} \sigma_i(L_G)$ to be 0. Then, the constraint $\text{rank}(L_G) = N - k$ in problem (13) would be satisfied.

Based on Ky Fan's Theorem [36], we have

$$\min \sum_{i=1}^{k} \sigma_i(L_G) = \min_{F\in R^{N\times k}, F^TF=I} Tr\left(F^T L_G F\right). \tag{14}$$

Hence, combining with the (14), we can further transform problem (13) into the following problem:

$$\begin{aligned} \min_{A,Z} \ & \|X - AZ\|_F^2 + \lambda Tr\left(F^T L_G F\right) \\ \text{s.t.} \ & Z^T\mathbf{1} = \mathbf{1}, \ Z \geq 0, F \in R^{N\times k}, \ F^TF = I. \end{aligned} \tag{15}$$

It was mentioned before that in order to avoid the trivial solution obtained by problem (8), we first select some sample points from the dataset to form the dictionary matrix. But it is not enough for this problem. This is because of the shortcoming in dictionary-based representation itself, that one point in a subspace can be linearly represented by points in that subspace and the points in other independent subspaces. So, to avoid this kind of trivial solution, we need to constrain the coefficient matrix Z. As the sparse constraint of SSC, we introduce a parameter c to constrain the coefficient matrix. Therefore, the final optimization problem can be rewritten as

$$\begin{aligned} \min_{A,Z,F} \ & \|X - AZ\|_F^2 + \lambda Tr\left(F^T L_G F\right) \\ \text{s.t.} \ & Z^T\mathbf{1} = \mathbf{1}, \ Z \geq 0, \ \|Z_i\|_0 \leq c, F \in R^{N\times k}, F^TF = I \end{aligned} \tag{16}$$

where Z_i represents the ith column vector of the coefficient matrix Z. The optimal problem (16) seems very complicated and not easy to obtain the optimal solution. In the next section, an alternation iteration-based technique is proposed to solve this problem.

IV. Optimization Strategy

In this section, we adopt an alternation iteration-based algorithm to solve the proposed model. For the optimization problem (16), there are three variables that need to be calculated, dictionary matrix A and coefficient matrices Z and F. Hence, we need to fix two of them to update the other one. The concrete derivation process is given next.

First, we fix matrices Z and F to solve the dictionary matrix A. Then, problem (16) becomes

$$\min_A \|X - AZ\|_F^2. \tag{17}$$

It is easy to see that problem (17) is a convex optimization problem. The solution can be obtained by taking the derivative as follows:

$$A = XZ^T\left(ZZ^T\right)^{-1}. \tag{18}$$

Hence, we obtain the optimal solution of subproblem (17) as shown in (18).

When matrices A and Z are fixed, problem (16) becomes

$$\begin{aligned} \min_F \ & Tr\left(F^T L_G F\right) \\ \text{s.t.} \ & F \in R^{N\times k}, \ F^TF = I. \end{aligned} \tag{19}$$

The optimal solution F is formed by the k eigenvectors of L_G corresponding to the k smallest eigenvalues.

When A and F are fixed, the optimization problem (16) becomes

$$\begin{aligned} \min_Z \ & \|X - AZ\|_F^2 + \lambda Tr\left(F^T L_G F\right) \\ \text{s.t.} \ & Z^T\mathbf{1} = \mathbf{1}, \ Z \geq 0, \ \|Z_i\|_0 \leq c. \end{aligned} \tag{20}$$

Subproblem (20) is still a difficult task to optimize. Therefore, we need to further handle this subproblem. Based on the property of the Laplacian matrix, we have the following formula:

$$Tr\left(F^T L_G F\right) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|\boldsymbol{f}^i - \boldsymbol{f}^j\right\|_2^2 G_{ij} \tag{21}$$

where $\boldsymbol{f}^i$ is the ith row of F.

Because the special structure of Laplacian matrix G defined in (10), (21) can be transformed into

$$Tr\left(F^T L_G F\right) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\boldsymbol{f}^i - \boldsymbol{f}^{j+n}\right\|_2^2 Z_{ij}. \tag{22}$$

Based on (22), problem (20) can be rewritten as

$$\begin{aligned}\min_{Z}\ & \sum_{i=1}^{n}\|X_i - AZ_i\|_2^2 + \lambda\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\boldsymbol{f}^i - \boldsymbol{f}^{j+n}\right\|_2^2 Z_{ij}.\\ \text{s.t.}\ & Z^T\mathbf{1} = \mathbf{1},\ Z \geq 0,\ \|Z_i\|_0 \leq c.\end{aligned} \tag{23}$$

We can see that the optimal solution Z_i from problem (23) is independent for each other. Hence, problem (23) can be turned into n subproblems and we can obtain the optimal solution individually. Denote the jth element of column vector $\boldsymbol{h}_i$ as $\|\boldsymbol{f}^i - \boldsymbol{f}^{j+n}\|_2^2$ where $i = 1, \ldots, n$ and $j = 1, \ldots, m$. Z_i also represents the ith column vector of coefficient matrix Z. Therefore, problem (23) is equivalent to optimize the following problem individually for each i:

$$\begin{aligned}\min_{Z_i}\ & \|X_i - AZ_i\|_2^2 + \lambda \boldsymbol{h}_i^T Z_i\\ \text{s.t.}\ & Z_i^T\mathbf{1} = 1,\ Z_i \geq 0,\ \|Z_i\|_0 \leq c.\end{aligned} \tag{24}$$

Taking the transformation for problem (24), the problem can be rewritten as

$$\begin{aligned}\min_{Z_i}\ & Z_i^T D Z_i + \boldsymbol{b}_i^T Z_i\\ \text{s.t.}\ & Z_i^T\mathbf{1} = 1,\ Z_i \geq 0,\ \|Z_i\|_0 \leq c\end{aligned} \tag{25}$$

here, D is a symmetric matrix denoted as $D = A^T A$, $\boldsymbol{b}_i \in R^{m\times 1}$ is a column vector denoted as $\boldsymbol{b}_i = \lambda \boldsymbol{h}_i - 2A^T X_i$.

Finally, we obtain the subproblem (25) which is easy to solve. A novel optimization algorithm that combines the augmented Lagrange method with the algorithm proposed by Kyrillidis *et al.* [37] is designed to deal with this problem. Next, we will give the details of the proposed algorithm to solve subproblem (25).

A. Suboptimization Strategy

In this section, we will propose a novel algorithm to solve the problem in the following form:

$$\begin{aligned}\min_{\boldsymbol{x}}\ & \boldsymbol{x}^T D\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}\\ \text{s.t.}\ & \boldsymbol{x}^T\mathbf{1} = 1,\ \boldsymbol{x} \geq 0,\ \|\boldsymbol{x}\|_0 \leq c\end{aligned} \tag{26}$$

here, $\boldsymbol{x}$ and $\boldsymbol{b}$ are both column vectors. By introducing a column vector $\boldsymbol{v}$ to problem (26), then the problem is equivalent to the following problem:

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{v}}\ & \boldsymbol{x}^T D\boldsymbol{v} + \boldsymbol{b}^T\boldsymbol{x}\\ \text{s.t.}\ & \boldsymbol{x}^T\mathbf{1} = 1,\ \boldsymbol{x} \geq 0,\ \|\boldsymbol{x}\|_0 \leq c,\ \boldsymbol{x} = \boldsymbol{v}.\end{aligned} \tag{27}$$

Based on problem (27), we have the augmented Lagrange function [38] of this problem as follows:

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{v}}\ & \boldsymbol{x}^T D\boldsymbol{v} + \boldsymbol{x}^T\boldsymbol{b} + \frac{\mu}{2}\left\|\boldsymbol{x} - \boldsymbol{v} + \frac{\mathbf{y}}{\mu}\right\|_2^2\\ \text{s.t.}\ & \boldsymbol{x}^T\mathbf{1} = 1,\ \boldsymbol{x} \geq 0,\ \|\boldsymbol{x}\|_0 \leq c\end{aligned} \tag{28}$$

where $\mathbf{y}$ is a Lagrange multiplier and μ is a penalty parameter.

When $\boldsymbol{x}$ is fixed, problem (28) becomes

$$\min_{\boldsymbol{v}} \boldsymbol{x}^T D\boldsymbol{v} + \frac{\mu}{2}\left\|\boldsymbol{x} - \boldsymbol{v} + \frac{\mathbf{y}}{\mu}\right\|_2^2. \tag{29}$$

Problem (29) is a convex optimization. Taking the derivative with respect to $\boldsymbol{v}$ and setting it to 0, we have the solution to this problem as follows:

$$\boldsymbol{v} = \frac{1}{\mu}\left(\mu\boldsymbol{x} + \mathbf{y} - D^T\boldsymbol{x}\right). \tag{30}$$

When $\boldsymbol{v}$ is fixed, problem (28) is equivalent to optimize the following problem:

$$\begin{aligned}\min_{\boldsymbol{x}}\ & \|\boldsymbol{x} - \boldsymbol{w}\|_2^2\\ \text{s.t.}\ & \boldsymbol{x}^T\mathbf{1} = 1,\ \boldsymbol{x} \geq 0,\ \|\boldsymbol{x}\|_0 \leq c\end{aligned} \tag{31}$$

here, $\boldsymbol{w} = \boldsymbol{v} - (1/\mu)\mathbf{y} - (1/\mu)D\boldsymbol{v} - (1/\mu)\boldsymbol{b}$. Problem (31) is easy to obtain the solution, Kyrillidis *et al.* [37] gave the analytical solution for this problem. For convenience, we introduce some additional notations. First, we denote $[w_i]_+ = \max(w_i, 0)$. Given a set $\mathcal{S} \in \mathcal{N} = \{1, \ldots, m\}$, the complement $\mathcal{S}^c$ is defined with respect to $\mathcal{N}$, and the cardinality is $\|\mathcal{S}\|$. The support set of $\boldsymbol{w}$ is $\text{supp}(\boldsymbol{w}) = \{i : w_i \neq 0\}$. Finally, we define that $\boldsymbol{w}_{|\mathcal{S}}$ is the vector where $\boldsymbol{w}$ is limited to $\mathcal{S}$ entries. Based on the above definitions, we can obtain the solution to problem (31) by

$$\boldsymbol{x}_{|\mathcal{S}^*} = \mathcal{P}_{\tilde{\lambda}+}\left(\boldsymbol{w}_{|\mathcal{S}^*}\right), \boldsymbol{x}_{|(\mathcal{S}^*)^c} = 0 \tag{32}$$

here, $\tilde{\lambda} = 1$ and $\mathcal{S}^* = \text{supp}(\mathcal{P}_{L_k}(\boldsymbol{w}))$. $\mathcal{P}_{L_k}(\boldsymbol{w})$ is an operator which keeps the c-largest entries of $\boldsymbol{w}$ and set the rest to 0. Besides, each entry of the vector $\mathcal{P}_{\tilde{\lambda}+}(\boldsymbol{w})$ is defined as follows:

$$\left(\mathcal{P}_{\tilde{\lambda}+}(\boldsymbol{w})\right)_i = [w_i - \tau]_+,\ \text{where}\ \tau := \frac{1}{\rho}\left(\sum_{i=1}^{\rho} w_i - \tilde{\lambda}\right)$$

for $\rho := \max\{j : w_j > (1/j)(\sum_{i=1}^{j} w_i - \tilde{\lambda})\}$. Hence, the procedure to solve problem (31) is summarized in Algorithm 1. As can be seen, the algorithm is very simple and concise.

Algorithm 1 Algorithm to Solve Problem (31)

1. Select support: $\mathcal{S}^* = supp(\mathcal{P}_{L_k}(\boldsymbol{w}))$.
2. Final solution: $\boldsymbol{x}_{|\mathcal{S}^*} = \mathcal{P}_{\tilde{\lambda}+}(\boldsymbol{w}_{|\mathcal{S}^*})$, $\boldsymbol{x}_{|(\mathcal{S}^*)^c} = 0$.

Therefore, in this section, we can design an algorithm to solve suboptimization problem (26) which can be seen from Algorithm 2.

Algorithm 2 Algorithm to Solve Problem (26)

Initialize: Let vector $\boldsymbol{x}$ be $\mathbf{0}$, $\boldsymbol{v} = \boldsymbol{x}$ and set the Lagrangian multiplier $\mathbf{y}$, parameter μ.

repeat

1. Fix $\boldsymbol{x}$, update the vector $\boldsymbol{v}$: based on the formula (30), $\boldsymbol{v} = \frac{1}{\mu}(\mu\boldsymbol{x} + \mathbf{y} - D^T\boldsymbol{x})$.
2. Fix $\boldsymbol{v}$, update the vector $\boldsymbol{x}$: when $\boldsymbol{v}$ is fixed, the solution $\boldsymbol{x}$ to problem (26) can be easily solved by Algorithm 1.
3: Update the multiplier $\mathbf{y}$ and μ: $\mathbf{y} = \mathbf{y} + \mu(\boldsymbol{x} - \boldsymbol{v})$, $\mu = \rho\mu$.

until converge

Algorithm 3 Algorithm to Solve Problem (16)

Input: data matrix $X \in R^{d\times n}$, the cluster number k, the dictionary size m and a large enough parameter λ.

Output: the learning coefficient matrix Z, G which is defined in Eq. (10) and the cluster label.

Initialize: dictionary matrix A, which is formed by m data samples in X. Randomly initialize the coefficient matrix Z to satisfy the constraint condition in problem (16).

repeat

1. Fix matrices X and Z, update A: based on Eq. (18)., we can get $A = XZ^T(ZZ^T)^{-1}$.
2. Fix matrices A and Z, update F: the matrix F is formed by k eigenvectors of $L_G = D_G - G$ corresponding to the k smallest eigenvalues.
3. Fix matrices A and F, update Z: For each column vector Z_i, optimize the problem (25) which can be solved by Algorithm 2.

until converge

Using k-means on the learning bipartite graph G to obtain the cluster label.

Based on the above conclusions, we have obtained the algorithm to solve subproblem (26). Hence, the entire process to solve the final objective problem can be concluded in Algorithm 3.

B. Convergence Analysis

For the proposed algorithm, we need to solve an inverse matrix problem, one nonconvex quadratic programming problem and one eigenvalue decomposition problem in each of the optimization iterations. The alternate iteration-based method is adopted in our algorithm. Therefore, when we optimize one variable with others fixed at each iteration, Algorithm 3 can be guaranteed to monotonically decrease the value of objective function (16). Meanwhile, the objective value is non-negative. As a consequence, Algorithm 3 can be guaranteed to converge to a local minimum of problem (16). At the same time, Algorithm 4 proposed in the next section also has the same property of convergence as Algorithm 3.

V. Algorithm Acceleration

From the previous section, the time complexity of our algorithm to solve problem (16) is a little high. Therefore, in this section, we optimize our algorithm to speed up the proposed technique. Combining the constraint condition with (10), the similarity G is non-negative. Hence, compared to Theorem 1, the normalized Laplacian matrix $\tilde{L}_G = I - D_G^{(1/2)} G D_G^{(1/2)}$ associated with G also has an important property, which can be seen in Theorem 2 [35].

Theorem 2: The multiplicity k of the eigenvalue 0 of the normalized Laplacian matrix $\tilde{L}_G$ is equal to the number of connected components in the graph associated with G.

Hence, based on Theorem 2, we know that if $\text{rank}(\tilde{L}_G) = N - k$, the constraint $G \in \Omega$ will be satisfied. So, problem (11) can be rewritten as

$$\begin{aligned} \min_{A,Z} \ & \|X - AZ\|_F^2 \\ \text{s.t.} \ & Z^T\mathbf{1} = \mathbf{1}, \ Z \geq 0, \ \text{rank}\left(\tilde{L}_G\right) = N - k. \end{aligned} \tag{33}$$

Similar to problem (12), we can convert the rank constraint condition into the form of problem (16). Finally, we can obtain the optimization problem as follows:

$$\begin{aligned} \min_{A,Z,F} \ & \|X - AZ\|_F^2 + \lambda Tr\left(F^T\tilde{L}_G F\right) \\ \text{s.t.} \ & Z^T\mathbf{1} = \mathbf{1}, \ Z \geq 0, \ \|Z_i\|_0 \leq c, \ F \in R^{N\times k}, \ F^TF = I. \end{aligned} \tag{34}$$

Next, we also propose an alternation iteration algorithm to optimize this proposed model.

When matrices Z and F are fixed, problem (34) can also be converted to the problem (17). Therefore, solution A can be obtained by (18).

Algorithm 4 Algorithm to Solve Problem (34)

Input: data matrix $X \in R^{d\times n}$, the cluster number k, the size of dictionary m and a large enough parameter λ.

Output: the learning coefficient matrix Z, G which is defined in Eq. (10) and the cluster label.

Initialize: dictionary matrix A, which is formed by m data points in X. Randomly initialize the coefficient matrix Z to satisfy the constraint condition in problem (34).

repeat

1. Fix matrices X and Z, update A: according to Eq. (18)., we can also get $A = XZ^T(ZZ^T)^{-1}$.
2. Fix matrices A and Z, update F: $F = \left[U^T \ V^T\right]^T$, the matrices U, V are the largest k left and right singular vectors of $D_{G_u}^{-\frac{1}{2}} Z D_{G_v}^{-\frac{1}{2}}$ respectively, and $D_G = \begin{bmatrix} D_{G_u} & \\ & D_{G_v} \end{bmatrix}$.
3. Fix matrices A and F, update Z: For each column vector Z_i, optimize the problem (40) which can be solved by Algorithm 2, where the j-th element of $\boldsymbol{h}_i'$ is $h_{ij}' = \|\frac{f^i}{\sqrt{d_i}} - \frac{f^{j+n}}{\sqrt{d_{j+n}}}\|_2^2$.

until converge

Using k-means on the learning bipartite graph G to obtain the cluster label.

When fixing Z and A, cause $\tilde{L}_G = I - D_G^{(1/2)} G D_G^{(1/2)}$, problem (34) is equivalent to the following problem:

$$\max_{F \in R^{N\times k}, F^T F = I} Tr\left(F^T D_G^{\frac{1}{2}} G D_G^{\frac{1}{2}} F\right). \tag{35}$$

F and D_G can be rewritten as a block form as follows:

$$F = \begin{bmatrix} U \\ V \end{bmatrix}, D_G = \begin{bmatrix} D_{G_u} & \\ & D_{G_v} \end{bmatrix}. \tag{36}$$

Because the block structure of similarity graph G, the problem (35) can be converted as

$$\max_{U^T U + V^T V = I} Tr\left(U^T D_{G_u}^{-\frac{1}{2}} Z D_{G_v}^{-\frac{1}{2}} V\right). \tag{37}$$

It is easy to see that problem (37) has the same form as problem (7) when $M = D_{G_u}^{-(1/2)} Z D_{G_v}^{-(1/2)}$. Therefore, based on Lemma 1, we have the optimal solutions U and V for this problem.

When matrices A and F are fixed, problem (34) can be rewritten as

$$\begin{aligned} \min_{A,Z,F} \ & \|X - AZ\|_F^2 + \lambda Tr\left(F^T \tilde{L}_G F\right) \\ \text{s.t.} \ & Z^T \mathbf{1} = \mathbf{1}, \ Z \geq 0, \ \|Z_i\|_0 \leq c. \end{aligned} \tag{38}$$

Due to the property of the normalized Laplacian matrix, we have

$$Tr\left(F^T \tilde{L}_G F\right) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left\| \frac{\boldsymbol{f}^i}{\sqrt{d_i}} - \frac{\boldsymbol{f}^j}{\sqrt{d_j}} \right\|_2^2 G_{ij}. \tag{39}$$

Here, d_i represents the ith diagonal element of matrix D_G. Therefore, combining the block structure of G defined in (10), and denote the jth element of column vector $\boldsymbol{h}_i'$ as $\|(\boldsymbol{f}^i/\sqrt{d_i}) - (\boldsymbol{f}^{j+n}/\sqrt{d_{j+n}})\|_2^2$. For each Z_i, problem (38) becomes the following same form as problem (25):

$$\begin{aligned} \min_{Z_i} \ & Z_i^T D Z_i + \boldsymbol{b}'^T_i Z_i \\ \text{s.t.} \ & Z_i^T \mathbf{1} = 1, \ Z_i \geq 0, \ \|Z_i\|_0 \leq c. \end{aligned} \tag{40}$$

Unlike problem (25), here, $\boldsymbol{b}_i' = \lambda \boldsymbol{h}_i' - 2A^T X_i$. We can also utilize Algorithm 2 to solve this problem. The detailed algorithm for solving problem (34) is summarized in Algorithm 4. In the algorithm, we only need to calculate SVD on the $m \times n$ matrix Z in each iteration. As a general rule, $\min(m, n) \ll (m + n)$, so Algorithm 4 is more efficient than Algorithm 3. Therefore, in the next section, we use Algorithm 4 to test the performance of the proposed model.

A. Complexity Analysis and Comparison

For Algorithm 4, in step 1, updating the dictionary A needs to compute the inverse of matrix ZZ^T, which takes $O(m^3)$. Step 2 only needs SVD of $D_{G_u}^{-(1/2)} Z D_{G_v}^{-(1/2)}$, so the complexity of step 2 is $O(nm^2)$. Step 3 needs to update each column of matrix Z, respectively, so it takes $O(nt)$ for obtaining Z, where t is the iteration number for each column vector Z_i. In summary, the time complexity of Algorithm 4 is $O(T(m^3 + nm^2 + nt))$, where T is the iteration number for the entire algorithm. The difference between Algorithms 3 and 4 is updating the matrix F in step 2. In Algorithm 3, it needs eigenvalue decomposition of L_G to obtain the eigenvectors, which takes $O((m+n)^3)$. Hence, the computational complexity of Algorithm 3 is $O(T(m^3 + (m+n)^3 + nt))$. It can be seen that Algorithm 4 is more efficient than Algorithm 3.

VI. Experiments

In this section, we will evaluate the performance of the proposed model (denoted by LAPIN) on three synthetic datasets and real-world data consisting of Hopkins 155 Datasets and four human face datasets. We also compare our method LAPIN with some typical methods and present the comparison results on benchmark datasets.

In some real-world datasets, due to noise and data dimension, we can learn an optimal bipartite graph with k connect components but the graph may not have a clear block-diagonal structure like Fig. 2(b), which means the results cannot be obtained from the learned graph straightly. So, we need to use k-means to handle this problem and obtain the final results.

A. Synthetic Data

First, we apply the proposed approach to three synthetic datasets as a sanity check, including two low-dimensional datasets (denoted by Synthetic-1 and Synthetic-2) and one high-dimensional dataset (denoted by Synthetic-3). The former two datasets are drawn from a 3-D (Here, the word 'dimensional' is abbreviated to 'D') Euclidean space, which consists of several groups distributed in low-dimensional subspaces such as one plane. Synthetic-1 is made up of one straight line that intersects one plane and Synthetic-2 consists of two intersecting planes. For the high-dimensional dataset Synthetic-3, we select three 40-D subspaces from the 100-D Euclidean space and take 100 samples from them, respectively, to build the datasets. These synthetic datasets are easy to construct and we utilize the basis vectors of subspaces to obtain these datasets.

For our algorithm, the dictionary learning mechanism can not only capture a better manifold structure for the coefficient matrix but also avoid the trivial solutions caused by the redundancy of the original datasets. Therefore, in this experiment, we randomly select 30% samples from original datasets as the dictionary initialization. Then, the proposed algorithm LAPIN is conducted ten times with different initializations on these datasets and the optimal solution that has the minimum objective value is selected as the final result. We set the parameters λ and c as 100 and 10, respectively, in this experiment.

The clustering results of Synthetic-1 and Synthetic-2 are shown in Fig. 1, different colors represent different clusters. We compare LAPIN with a classical method BDR proposed by Lu *et al.* [24], which also aims to learn a block structure for the graph. From Fig. 1(a) and (b), the clustering accuracies obtained by BDR on Synthetic-1 and Synthetic-2 datasets are 55.50% and 73.13%, respectively. However, our model LAPIN achieves the accuracy of 95.00% and 94.38% on these two datasets based on Fig. 1(c) and (d). Therefore, It can be seen that LAPIN is better at capturing the local structures in the low-dimensional space. Because our algorithm

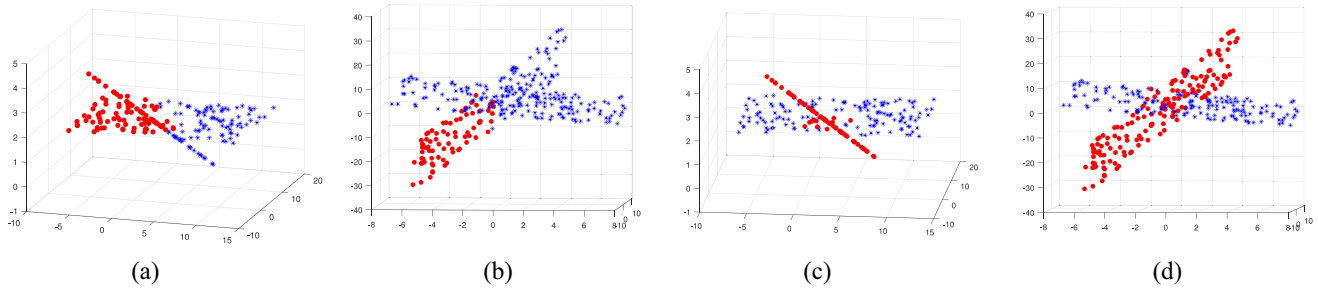

Fig. 1. Clustering results on two toy datasets Synthetic-1 and Synthetic-2 by the proposed algorithm LAPIN and the state-of-the-art method BDR. (a) Result on Synthetic-1 by BDR (55.50%). (b) Result by BDR on Synthetic-2 (73.13%). (c) Result of LAPIN on Synthetic-1, and the accuracy is 95.00%. (d) Result on Synthetic-1 by LAPIN (94.38%).

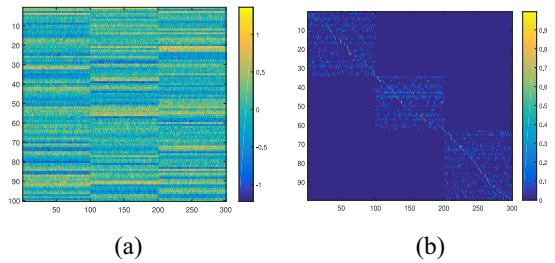

Fig. 2. Clustering results on high-dimensional dataset Synthetic-3. (a) Graph generated from the original dataset. (b) Learned structure of coefficient matrix Z by LAPIN.

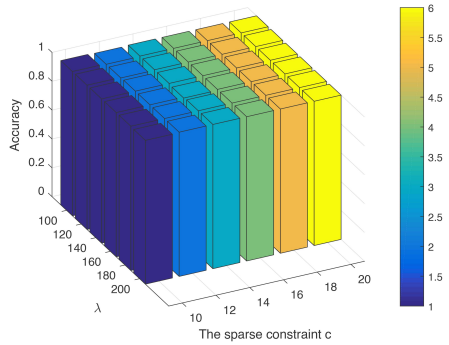

Fig. 3. Clustering results with different values of two parameters λ and c on high-dimensional dataset Synthetic-3 by the proposed algorithm LAPIN.

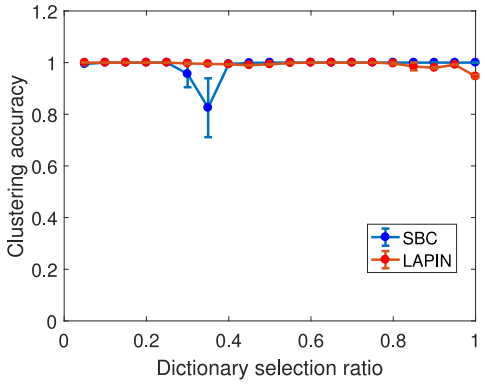

Fig. 4. Clustering results on Synthetic-3 with different selection ratios of the dictionary. The red and blue lines correspond to the change diagrams of clustering accuracy obtained by LAPIN and SBC.

can uniform the dictionary learning mechanism and sparse constraint effectively to avoid the trivial solutions caused by the redundancy of the dataset. The traditional dictionary-based methods, such as BDR, cannot handle the low-dimensional datasets very well. Because they utilize the original dataset as the dictionary, which leads to the redundancy problem of the dictionary.

For Synthetic-3, we show the graph of the original dataset and the learned structure of coefficient matrix Z by LAPIN in Fig. 2. It can be seen that the learned structure of Z has a perfect block-diagonal structure and the clustering accuracy on this high-dimensional dataset is 100%. Therefore, based on the rank constraint and dictionary learning mechanism, our model can capture the optimal structure from the high-dimensional dataset to achieve better performance. It further demonstrates that our algorithm not only handles the low-dimensional datasets very well but also has good performance on high-dimensional datasets.

Furthermore, in order to verify the robustness of our model, the sensitivity of parameters λ and c is tested in this experiment. We conduct LAPIN on the Synthetic-3 with different values of parameters λ and c and obtain the clustering precision as shown in Fig. 3. As we can see, there exists little difference in clustering results with the two parameters λ and c scanned in the entire range, and the results are all at a high precision level. Besides, when the value of parameter λ is increasing, the performance of LAPIN will increase effectively.

In addition, to verify the influence of the dictionary's size on the clustering result, we test our algorithm on the Synthetic-3 dataset with different dictionary selection ratios. The selection ratios range from 5% to 100%, which can be seen in Fig. 4. Besides, we compare LAPIN with a dictionary-based method subspace biclustering (SBC) [39], which is also based on dictionary learning. As can be seen from Fig. 4, there is not much difference between these two algorithms in the selection ratios of the dictionary, and both of them have stable clustering performance. It illustrates that for the small datasets with no noise, samples in each subspace can be well represented by the elements of the updated dictionary so that the optimal coefficient matrix can be obtained. To sum up, according to Figs. 3 and 4, our algorithm LAPIN is robust within a certain

TABLE I
INTRODUCTION OF THE HOPKINS 155 DATASET

Category	Checkerboard	Traffic	Others	All
# Seq.	78	31	11	120
Points	291	241	155	266
Frames	28	30	40	30
# Seq.	26	7	2	35
Points	437	332	122	398
Frames	28	31	31	29

parameter range. Therefore, in the experiments of the next section, we set the parameters λ and c and the selection ratio of the dictionary in the proposed algorithm to 100, 10, and 30%, respectively.

B. Real-World Data

In experiments with real data, we, respectively, verify the validity of LAPIN in two aspects of motion segmentation and human face clustering. We will introduce the experiment setting and present the clustering results on benchmark datasets.

Motion Segmentation: The Hopkins 155 dataset is a motion segmentation dataset and provided by the Vision Lab of Johns Hopkins University. This dataset consists of 155 sequences and can be divided into three main categories: 1) checkerboard; 2) traffic; and 3) other sequence (Articulated/nonrigid). The 120 sequence contains two motions (i.e., clusters) and the rest have three motions. Each sequence is one clustering task that needs to be segmented into two or three motions. More details of the Hopkins 155 dataset can be seen in Table I. Some samples from Hopkins 155 are shown in Fig. 5.

Therefore, in this experiment, we have 155 clustering tasks. For each sequence, the feature points and sample points of its dataset are different. In order to handle 155 sequences uniformly and easily, based on the [24], we utilize the PCA [40] as the preprocessing step to reduce the dimension of each sequence and save the 95% information of the original dataset.

We compared the proposed model LAPIN with other classical methods, including random sample consensus (RANSAC) [41], local subspace analysis (LSA) [42], SSC [22], SSC-OMP [43], LRR (Robust Low-Rank Representation) [23], LatLRR [44], LRS (Subspace Clustering via New Low-Rank Model) [33], BDR (Subspace Clustering by Block-Diagonal Representation) [24], and SBC [39]. These algorithms are tested on the same datasets that have been preprocessed by PCA based on the method mentioned in [24]. Besides, the parameters in all algorithms are tuned to the best. Furthermore, the low-rank SSC algorithms [45], including GMC-LRSSC and S_0/ℓ_0-LRSSC are introduced in this experiment for comparison with our method LAPIN. Based on the parameter setting in [45], we retained the values of parameters with the best performance after several experiments. In this section, we set the parameters $\alpha = 1$, $\mu_2^0 = 1$, and $\gamma = 0.1$ for GMC-LRSSC and $\lambda = 0.5$ and $\mu^0 = 20$ for S_0/ℓ_0-LRSSC. Table II shows the comparison results of LAPIN with other methods.

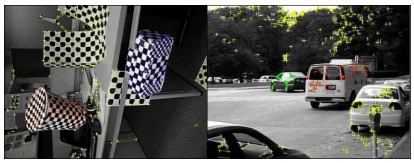

Fig. 5. Some image samples from Hopkins 155.

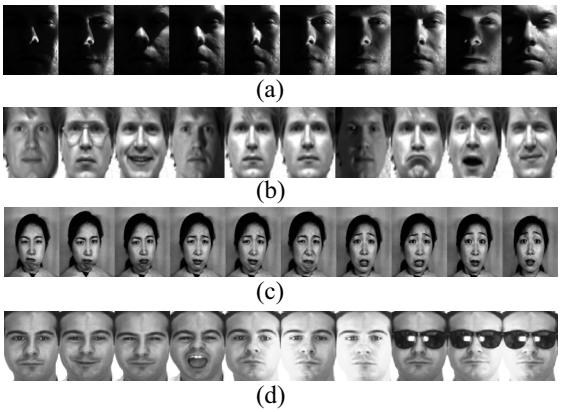

Fig. 6. Some image samples from four benchmark datasets. (a) Samples from Extended Yale Database B image. (b) Yale Face Database image samples. (c) JAFFE face samples. (d) AR image samples.

Table II gives the clustering results for all algorithms, including mean accuracies, median accuracies, minimum accuracies, and standard errors. The best and second-best results are given in bold and underlined. It can be seen that our method LAPIN can achieve almost the same clustering performance as the best method BDR. In particular, our model LAPIN obtains the best experimental results on minimum accuracies and standard errors in two motions. It confirms that LAPIN has higher stability in processing this motion segmentation dataset than the comparable algorithms.

Image Classification on Four Real-World Datasets: In this section, we conduct LAPIN on four human face datasets, including: 1) the Extended Yale B dataset [46]; 2) Yale Face Dataset [47]; 3) JAFFE [48]; and 4) AR [49]. Some samples from the four human face datasets are presented in Fig. 6. Besides, two handwritten digital datasets: 1) MNIST_10 [50] and 2) USPS [51] are introduced in this experiment. To further clarify these datasets, we will give an introduction for them in detail.

Extended Yale B has different face images of 38 persons and each person owns 64 near frontal images under different illuminations and poses. The Yale Face Dataset provided by the YALE Center for Computational Vision and Control consists of grayscale images of 15 individuals and each individual has 165 images. The size of each image is 112×92 and these images are taken under different lighting, different expression, and with or without glasses. There are ten Japanese female models in the JAFFE dataset and each female contains 213 images of seven facial expressions (six basic facial expressions and one neutral). The AR face dataset has over 4000 color face images and the size of each image is 165×120. The AR dataset contains 126 classes, which are 70 males and

TABLE II
CLUSTERING ACCURACY (%) AND STANDARD ERROR (%) ON THE HOPKINS 155

Method	two motions				three motions				All			
	mean	median	min	std	mean	median	min	std	mean	median	min	std
RANSAC	94.88	98.63	54.33	9.38	81.51	86.18	52.55	14.26	91.86	97.22	52.55	12.01
LSA	95.29	99.56	51.59	10.91	90.57	95.86	46.59	14.35	94.22	99.33	46.59	11.89
SSC	92.93	99.29	53.46	12.79	80.45	88.27	47.34	15.69	90.11	97.45	47.34	14.43
SSC-OMP	67.61	65.25	50.33	10.71	53.23	53.38	34.30	10.29	64.36	63.00	36.30	12.81
LRR	97.32	**100.00**	53.53	7.13	88.29	93.50	58.20	14.06	95.28	**100.00**	53.53	9.86
LatLRR	97.89	**100.00**	55.35	7.71	96.26	**100.00**	64.09	8.58	97.52	**100.00**	55.35	7.92
LRS	96.68	**100.00**	67.15	6.89	87.60	89.15	67.77	9.79	94.63	99.48	67.15	8.51
BDR	**98.74**	**100.00**	51.52	6.46	**98.78**	99.79	**87.59**	**2.50**	**98.74**	**100.00**	51.52	5.80
SBC	74.63	73.49	52.38	11.62	62.49	59.65	38.83	11.98	71.74	71.55	38.83	12.83
GMC-LRSSC	85.11	88.07	56.04	7.63	78.25	79.82	56.69	10.22	83.61	85.28	56.04	8.51
S_0/ℓ_0-LRSSC	81.79	83.29	54.92	10.88	73.46	73.66	54.68	11.71	79.99	80.92	54.68	11.07
LAPIN	98.50	**100.00**	**83.56**	**2.79**	96.52	98.37	74.95	5.63	98.05	99.94	**74.95**	**3.70**

TABLE III
COMPARISONS WITH CLUSTERING ACCURACY (ACC) (%) AND NMI (%) ON THE PRESENTED IMAGE DATASETS

Method	Extended Yale B		Yale Face		JAFFE		AR		MNIST_10		USPS	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
RANSAC	20.66	10.49	27.68	26.32	31.55	30.85	22.98	15.04	17.62	19.08	39.34	32.09
LSA	58.26	57.03	70.77	67.09	65.54	68.00	43.08	44.71	16.84	17.58	21.18	20.76
SSC	58.75	57.63	69.77	70.30	78.19	79.24	59.69	61.37	51.87	46.96	50.84	56.24
SSC-OMP	79.29	80.67	45.05	48.33	24.69	21.41	44.58	47.57	43.32	42.97	20.83	21.51
LRR	86.89	82.13	69.13	71.44	62.44	70.83	63.81	65.63	52.43	48.83	68.43	62.33
LatLRR	79.81	75.71	69.09	70.73	60.68	70.16	63.85	65.66	51.46	48.46	68.56	62.54
LRS	75.16	83.13	50.91	51.81	67.61	71.68	55.00	56.90	**60.41**	54.10	69.68	67.59
BDR	93.91	90.36	39.09	42.67	62.44	74.69	60.00	60.52	52.40	46.69	70.03	64.53
SBC	77.81	71.37	48.18	43.12	43.66	37.35	37.69	32.30	21.32	19.36	36.79	33.16
GMC-LRSSC	94.84	89.65	36.36	33.80	60.56	63.04	63.08	65.46	50.37	42.09	69.93	63.29
S_0/ℓ_0-LRSSC	**95.78**	**91.77**	47.27	42.70	70.42	72.45	67.31	69.17	49.99	44.56	61.36	57.17
LAPIN	88.12	82.89	**81.45**	**79.42**	**80.75**	**82.89**	**73.08**	**69.36**	60.29	**56.72**	**71.69**	**68.28**

56 females, respectively. For MNIST and USPS, they belong to the grayscale images of handwritten digits and both have ten clusters associated with the digital set $\{0, 1, \ldots, 9\}$. The MNIST dataset consists of 70 000 samples with the size of 28×28, we randomly choose 10% samples in each cluster to build the experimental dataset MNIST_10. The image size of USPS is 16×16, and it is composed of 9298 samples in total.

In this experiment, we all select the first ten classes as the experimental dataset for these benchmark datasets. The same preprocessing method as the motion segmentation is used to reduce the dimension of each image dataset, which keeps the 95% information for the original dataset. We present the experimental metrics, including cluster accuracy (ACC) and normalized mutual information (NMI) as the standard for measuring the algorithms' performance. We also compare LAPIN with other classical subspace clustering models, which are RANSAC [41], LSA [42], SSC [22], SSC-OMP [43], Robust LRR [23], LatLRR [44], Subspace Clustering via New low-rank model (LRS) [33], subspace clustering by BDR [24], and SBC [39]. Like the motion segmentation, the parameters of these methods are tuned to the best of themselves. Based on the face recognition experiment in [45], we set $\alpha = 1000$, $\mu_2^0 = 3$, and $\gamma = 1$ for GMC-LRSSC and $\lambda = 0.5$ and $\mu^0 = 1$ for S_0/ℓ_0-LRSSC in this section.

Table III gives the final results for all compared methods with the proposed model LAPIN, the bold and underlined data represent the best and second-best results in this experiment, respectively. We can see that our model has higher accuracy and NMI on Yale Face, JAFFE, AR, and USPS datasets. In the Extended Yale B dataset, BDR, GMC-LRSSC, and S_0/ℓ_0-LRSSC all obtain better results in ACC and NMI than our method LAPIN, but they do not perform consistently on other benchmark datasets. On the contrary, LAPIN is better at handling all of these datasets and the clustering accuracy obtained on Extended Yale B is next to BDR, GMC-LRSSC, and S_0/ℓ_0-LRSSC. It demonstrates that LAPIN is more capable of handling real datasets compared to other state-of-the-art methods.

According to Tables II and III, the dictionary-based method SBC does not perform well on the Hopkins 155 dataset and the presented image datasets. Compared to the proposed algorithm LAPIN, there are two main reasons for this problem. One is that SBC divides the clustering process into two steps: 1) dictionary learning and 2) graph clustering, which makes the updated dictionary not necessarily applicable to the graph clustering process in some datasets. The other is that the learned graph by SBC may not have the same block structure as in LAPIN, because SBC does not impose the rank constraint of the Laplacian matrix on the learned graph. Therefore, SBC

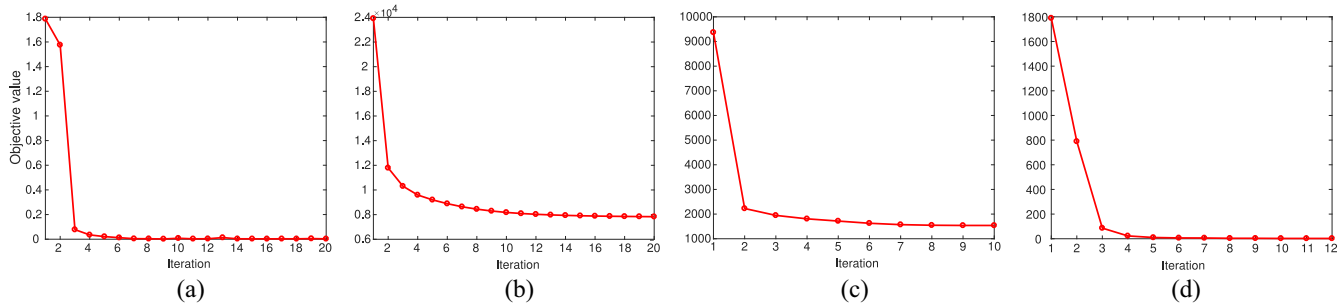

Fig. 7. Convergence of our algorithm LAPIN on four image databases. (a) Extended Yale Database B image dataset. (b) Yale face database. (c) JAFFE face database. (d) AR image dataset. According to these figures, we can find our algorithm converges directly from 5–15 iterations in these four image databases.

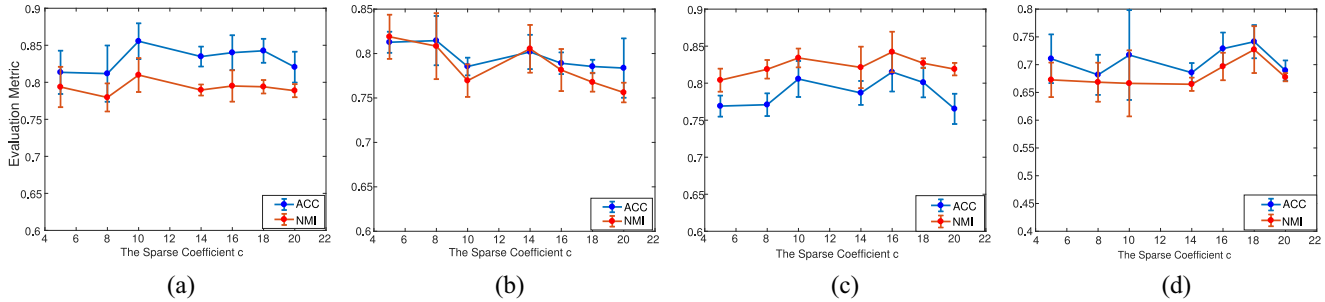

Fig. 8. Clustering results by the proposed algorithm LAPIN on four human face image datasets with different values of sparse coefficient c, which has a range of [5, 20]. The curves of ACC and NMI for each dataset are placed in the subgraph of Fig. 8, the red line in each figure represents NMI and the blue one represents the clustering accuracy (ACC). These figures demonstrate that LAPIN has a very stable clustering performance on each dataset with the different values of parameter c. (a) Extended Yale B. (b) Yale face. (c) JAFFE. (d) AR.

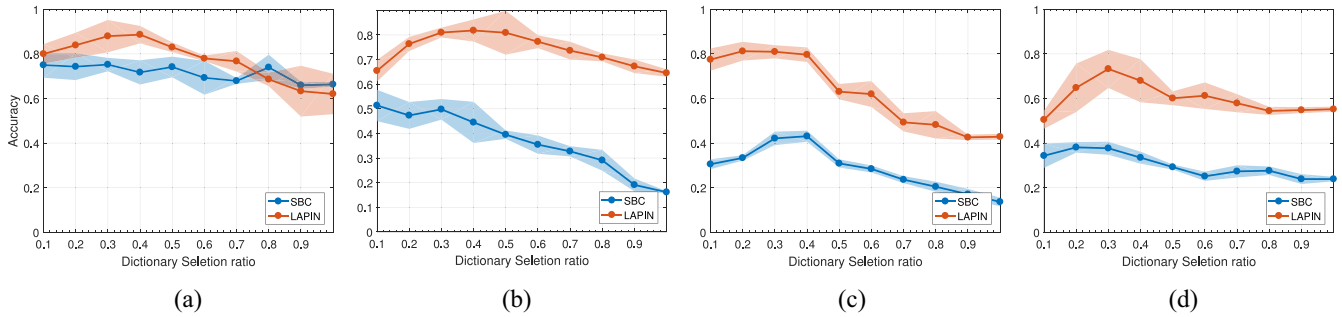

Fig. 9. Clustering accuracies on four human face image datasets with different selection ratios for dictionary. The change curves obtained by LAPIN and SBC are, respectively, represented by the red and blue lines. Besides, the shaded area represents the size of the standard deviation. From these figures, we can conclude that both LAPIN and SBC show stable and good performance when the dictionary selection ratio ranges from 20% to 40%. (a) Extended Yale B. (b) Yale face. (c) JAFFE. (d) AR.

cannot perform better on these benchmark datasets than our algorithm LAPIN.

The convergence curves of the proposed algorithm LAPIN on these four human face datasets are also given in Fig. 7. We can see that our algorithm can converge after 5–15 iterations when conducting on these benchmark datasets. It means LAPIN is capable of dealing with the large-scale dataset when the dimensionality is not too high. Besides, we also verify the sensitivity of sparse coefficient c in this experiment. The proposed algorithm LAPIN is conducted on these four image datasets and we obtain the final clustering results, including accuracies (ACC) and NMI, which can be seen in Fig. 8. It shows that LAPIN has the ability to handle these benchmark datasets and achieves excellent performance with different values of parameter c ranging from 5 to 20. Furthermore, it demonstrates that the sparse coefficient c has little effect on the performance of the proposed algorithm LAPIN.

Fig. 9 presents the sensitivity analysis results of the dictionary selection ratio on four human face image datasets, which are obtained by our model LAPIN and the dictionary-based method SBC. The selection ratio ranges from 10% to 100% and the change curves obtained by LAPIN and SBC are, respectively, represented by red and blue lines. Compared to the result of the synthetic dataset shown in Fig. 8, we can conclude that the problem of noise and dictionary redundancy indeed exists in the real scenario. It is very necessary to select and update the dictionary for subspace clustering methods. Fig. 9 shows that both LAPIN and SBC perform very well and stably when setting the range of selection ratio to be [20%, 40%]. Therefore, in this article, the dictionary selection

ratio is set as 30% for the proposed model LAPIN to compare with other SOTA methods.

VII. Conclusion

This article focused on addressing the problem of optimal graph construction for subspace clustering. By utilizing the relationship between the graph and the rank of its Laplacian matrix, we proposed a representation-based model with a bipartite graph that can be learned to have exactly k connect components. Because the learned graph contains the duality information between samples and features, our model has a better performance than other graph-based methods that only utilize the relationships between samples. In this article, we utilized partial samples to initialize the dictionary instead of the original data and update the dictionary in the algorithm to avoid the trivial solution caused by redundant information hiding in the data. Besides, we introduced the sparse coefficient c in our model, which makes LAPIN more robust to the low-dimensional dataset as shown in Fig. 1. The experimental results on both synthetic and benchmark datasets demonstrated the effectiveness and stability of our model LAPIN.

However, though LAPIN can find the graph with exactly k connect components, the learned structure may not have the clear block diagonal as shown in Fig. 2 due to the noise and data dimension in some real-world datasets. Hence, we have to use the k-means technique as the postprocessing to obtain the final results, which does not reach our ultimate goal. So, there are still some interesting works to study in the future. In addition, it would be useful to modify the proposed LAPIN handling the noise with the method LRR by Lin *et al.* [23]. That can make LAPIN not only obtain the clustering results from the learned graph straight but also more robust to noise.

References

[1] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, 2001, pp. 383–390.

[2] A. Zhang, N. Fawaz, S. Ioannidis, and A. Montanari, "Guess who rated this movie: Identifying users through subspace clustering," in *Proc. 28th Conf. Uncertainty Artif. Intell.*, 2012, pp. 944–953.

[3] N. M. Nasrabadi, "Pattern recognition and machine learning," *J. Electron. Imag.*, vol. 16, no. 4, 2007, Art. no. 049901.

[4] X. Wang, Z. Lei, X. Guo, C. Zhang, H. Shi, and S. Z. Li, "Multi-view subspace clustering with intactness-aware similarity," *Pattern Recognit.*, vol. 88, pp. 50–63, Apr. 2019.

[5] W. Hong, J. Wright, K. Huang, and Y. Ma, "Multiscale hybrid linear models for lossy image representation," *IEEE Trans. Image Process.*, vol. 15, pp. 3655–3671, 2006.

[6] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Comput. Vis. Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[7] X. Yang, L. Liao, Q. Yang, B. Sun, and J. Xi, "Limited-energy output formation for multiagent systems with intermittent interactions," *J. Franklin Inst.*, vol. 358, no. 13, pp. 6462–6489, 2021.

[8] T. Hastie and P. Y. Simard, "Metrics and models for handwritten character recognition," *Stat. Sci.*, vol. 13, no. 1, pp. 54–65, 1998.

[9] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using powerfactorization and GPCA," *Int. J. Comput. Vis.*, vol. 79, no. 1, pp. 85–105, 2008.

[10] C. Tang *et al.*, "Feature selective projection with low-rank embedding and dual Laplacian regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1747–1760, Sep. 2019.

[11] L. Tian, F. Nie, R. Wang, and X. Li, "Learning feature sparse principal subspace," in *Advances in Neural Information Processing Systems*, vol. 33. Red Hook, NY, USA: Curran Assoc., 2020.

[12] C. Tang *et al.*, "CGD: Multi-view clustering via cross-view graph diffusion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 5924–5931.

[13] J. Ma, Y. Zhang, and L. Zhang, "Discriminative subspace matrix factorization for multiview data clustering," *Pattern Recognit.*, vol. 111, Mar. 2021, Art. no. 107676.

[14] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum, "Estimation of subspace arrangements with applications in modeling and segmenting mixed data," *SIAM Rev.*, vol. 50, no. 3, pp. 413–458, 2008.

[15] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, Dec. 2005.

[16] A. Gruber and Y. Weiss, "Multibody factorization with uncertainty and missing data using the EM algorithm," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2004, pp. 707–714.

[17] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, "Hyperspectral image unsupervised classification by robust manifold matrix factorization," *Inf. Sci.*, vol. 485, pp. 154–169, Jun. 2019.

[18] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, 1999.

[19] Y. Sugaya and K. Kanatani, "Geometric structure of degeneracy for multi-body motion segmentation," in *Proc. Int. Workshop Stat. Methods Video Process.*, 2004, pp. 13–25.

[20] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2017, pp. 4129–4138.

[21] C. Tang *et al.*, "Learning a joint affinity graph for multiview subspace clustering," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1724–1736, Jul. 2019.

[22] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[23] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 663–670.

[24] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 487–501, Feb. 2019.

[25] W. Chang, F. Nie, R. Wang, and X. Li, "Robust subspace clustering by learning an optimal structured bipartite graph via low-rank representation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 3692–3696.

[26] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[27] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[28] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[29] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, p. 717, 2009.

[30] R. H. Keshavan, S. Oh, and A. Montanari, "Matrix completion from a few entries," in *Proc. IEEE Int. Symp. Inf. Theory*, 2009, pp. 324–328.

[31] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[32] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2001, pp. 269–274.

[33] F. Nie and H. Huang, "Subspace clustering via new low-rank model with discrete group structure constraint," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1874–1880.

[34] F. R. Chung and F. C. Graham, *Spectral Graph Theory*. Providence, RI, USA: Amer. Math. Soc., 1997.

[35] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The Laplacian spectrum of graphs," in *Graph Theory, Combinatorics, and Applications*, vol. 2. New York, NY, USA: Wiley, 1991, p. 12.

[36] K. Fan, "On a theorem of Weyl concerning eigenvalues of linear transformations I," *Proc. Nat. Acad. Sci.*, vol. 35, no. 11, pp. 652–655, 1949.

[37] A. Kyrillidis, S. Becker, V. Cevher, and C. Koch, "Sparse projections onto the simplex," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 235–243.

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

[39] A. Adler, M. Elad, and Y. Hel-Or, "Linear-time subspace clustering via bipartite graph modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2234–2246, Oct. 2015.
[40] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometr. Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[41] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[42] J. Yan and M. Pollefeys, *A General Framework for Motion Segmentation: Independent, Articulated, Rigid, Non-Rigid, Degenerate and Non-Degenerate*, Springer, 2006, pp. 94–106.
[43] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3918–3927.
[44] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1615–1622.
[45] M. Brbić and I. Kopriva, "ℓ_0-motivated low-rank sparse subspace clustering," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1711–1725, Apr. 2020.
[46] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.
[47] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
[48] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proc. 3rd IEEE Int. Conf. Autom. Face Gesture Recognit.*, 1998, pp. 200–205.
[49] A. M. Martinez, "The AR face database," Dept. Comput. Sci., Universitat Autònoma Barcelona, Bellaterra, Spain, Rep. CVC #24, 1998.
[50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[51] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

Feiping Nie (Senior Member, IEEE) received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2009.

He is currently a Full Professor with Northwestern Polytechnical University, Xi'an, China. He has published more than 100 papers in the following journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *International Journal of Computer Vision*, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, and ACM MM. His papers have been cited more than 20 000 times and the H-index is 78. His research interests are machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval.

Prof. Nie is currently serving as an associate editor or a PC member for several prestigious journals and conferences in the related fields.

Wei Chang received the master's degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2020, where he is currently pursing the Doctoral degree with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics.

His research interests are mainly about the unsupervised learning and multitask learning in machine learning, computer vision, and data mining.

Rong Wang received the B.S. degree in information engineering, the M.S. degree in signal and information processing, and the Ph.D. degree in computer science from Xi'an Research Institute of Hi-Tech, Xi'an, China, in 2004, 2007, and 2013, respectively.

From 2007 to 2013, he also studied with the Department of Automation, Tsinghua University, Beijing, China, for his Ph.D. degree. He is currently an Associate Professor with the School of Cybersecurity and the School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University, Xi'an. His research interests focus on machine learning and its applications.

Xuelong Li (Fellow, IEEE) is a Full Professor with the School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University, Xi'an, China.