Scalable Graph-Based Clustering With Nonnegative Relaxation for Large Hyperspectral Image

Rong Wang, *Member, IEEE*, Feiping Nie, *Member, IEEE*, Zhen Wang, Fang He, and Xuelong Li, *Fellow, IEEE*

***Abstract*— Hyperspectral image (HSI) clustering is very important in remote sensing applications. However, most graph-based clustering models are not suitable for dealing with large HSI due to their computational bottlenecks: the construction of the similarity matrix W, the eigenvalue decomposition of the graph Laplacian matrix L, and k-means or other discretization procedures. To solve this problem, we propose a novel approach, scalable graph-based clustering with nonnegative relaxation (SGCNR), to cluster the large HSI. The proposed SGCNR algorithm first constructs an anchor graph and then adds the nonnegative relaxation term. With this, the computational complexity can be reduced to $O(nd\log m + nK^2 + nKc + K^3)$, compared with traditional graph-based clustering algorithms that need at least $O(n^2d + n^2K)$ or $O(n^2d + n^3)$, where n, d, m, K, and c are, respectively, the number of samples, features, anchors, classes, and nearest neighbors. In addition, the SGCNR algorithm can directly obtain the clustering indicators, without resort to k-means or other discretization procedures as traditional graph-based clustering algorithms have to do. Experimental results on several HSI data sets have demonstrated the efficiency and effectiveness of the proposed SGCNR algorithm.**

***Index Terms*— Anchor graph, graph-based clustering, hyperspectral image (HSI), nonnegative relaxation.**

I. Introduction

By combining imaging technology with spectroscopy technology, hyperspectral remote sensing obtains spatial and spectral continuous hyperspectral image (HSI) data [1]–[5]. It provides rich spectral and spatial information for monitoring the earth's surface and fine identification of various land cover materials. Therefore, HSI is widely used in many remote sensing applications such as precision agriculture, mineral exploration, disaster monitoring, and ecological environment monitoring [6]–[9]. In these applications, clustering is a basic technique commonly used in HSI processing [10]. The aim of HSI clustering is to partition a given image into groups such that pixels in the same group are as similar to each other as possible, while those assigned to different groups are dissimilar. Due to the intrinsic complexity of the distribution of the ground objects and the large spectral variability and complex spatial structures, clustering has always been one of the most challenging tasks in HSI processing [11]–[13].

Up until now, many clustering methods of various working mechanisms have been proposed for HSI processing. Zhang *et al*. [3] summarized the existing HSI clustering algorithms and broadly divided them into the following four categories: the centroid-based, density-based, biological-based, and graph-based clustering algorithms.

(i) The centroid-based clustering algorithms, such as k-means [14], fuzzy c-means (FCM) [15] and FCM_S1 [16], and kernel-based fuzzy clustering algorithm [17], cluster the HSIs based on the similarity measure. Euclidian distance is the most commonly used similarity measure in these algorithms. Nevertheless, they belong to mountain-climbing methods that easily get stuck in a local optimum, and cannot always be sure to work well due to their sensitivity to initialization and noise. In addition, they need the data to satisfy the hypothesis of Gaussian distribution, which is usually not the case of HSIs.

(ii) The density-based clustering algorithms, such as density-based spatial clustering algorithm with noise (DBSCAN) [18] and clustering by fast search and find of density peaks (CFSFDP) [19], form the clusters by directly searching for connected dense regions in the feature space, where clusters are defined as high-density regions in the feature space separated by low-density regions. Different algorithms use different definitions of connectedness. Although they are attractive because of their inherent ability to deal with arbitrary shaped clusters, they have limitations in handling HSI which belong to the high-dimensional data. The feature space of high-dimensional data is usually sparse, making it difficult to distinguish high-density regions from low-density regions.

(iii) The biological-based clustering algorithms, such as remote sensing unsupervised artificial immune network

Manuscript received December 10, 2018; revised February 21, 2019; accepted April 18, 2019. Date of publication May 14, 2019; date of current version September 25, 2019. This work was supported by the National Natural Science Foundation of China under Grant U1803263, Grant 61772427, Grant 61751202, and Grant 61871470. *(Corresponding author: Feiping Nie.)*

R. Wang is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, X'ian 710072, China (e-mail: wangrong07@tsinghua.org.cn).

F. Nie and X. Li are with the School of Computer Science and the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: feipingnie@gmail.com; li@nwpu.edu.cn).

Z. Wang is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China, and also with the School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: zhenwang0@gmail.com).

F. He is with the Xi'an Research Institute of Hi-Tech, Xi'an 710025, China (e-mail: fanghe1107@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2019.2913004

(RSUAIN) [20], automatic fuzzy clustering method based on adaptive multiobjective differential evolution (AFCMDE) [21], adaptive memetic fuzzy clustering algorithm with spatial information for remote sensing imagery (AMASFC) [7], and adaptive multiobjective memetic fuzzy clustering algorithm (AFCMOMA) [22], cluster the HSIs by utilizing the biological models. However, these algorithms cannot always obtain satisfactory clustering results due to the biological models do not always accord with the characteristics of HSIs.

(iv) The graph-based clustering algorithms, such as spectral clustering (SC) [23], Ratio Cut [24], Normalized Cut [25], SC clustering (SCC) [26], and sparse subspace clustering (SSC) [3], [12], [13], [27], all follow the same scheme that has three key parts. First, it computes the weights that model the similarity between pairs of data points and gives rise to the similarity matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$, where n denotes the number of data points. Second, it computes K eigenvectors of the corresponding graph Laplacian matrix $\boldsymbol{\mathcal{L}}$, where K is the number of clusters. Finally, k-means or other discretization procedures is needed to uncover the clustering indicators. They root in exploiting the nonlinear pairwise similarity between data instances and have also shown great potential in HSI clustering [3], [12], [13], [27].

In this paper, we focus on the family of graph-based clustering algorithms that have achieved a significant advantage in many real applications. However, the traditional graph-based clustering algorithms are typically not the first choice for large HSIs clustering problem since modern HSI data sets exhibit great challenges in both computation and memory consumption for constructing the similarity matrix $\boldsymbol{W}$ and computing K eigenvectors of the corresponding graph Laplacian matrix $\boldsymbol{\mathcal{L}}$ [4]. Given a data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ whose underlying data distribution can be represented as K weakly inter-connected clusters. First, they need $O(n^2)$ space to store the similarity matrix $\boldsymbol{W}$ and $O(n^2 d)$ complexity to construct the similarity matrix $\boldsymbol{W}$, where n and d are the number of samples and features, respectively. Then, they at least take $O(n^2 K)$ or $O(n^3)$ complexity to compute K eigenvectors of the graph Laplacian matrix $\boldsymbol{\mathcal{L}}$, depending on whether iterative or a direct eigensolver is used. In addition, they usually need to perform other clustering methods, such as k-means, on the obtained eigenvectors to get the final clustering results. Altogether, graph-based clustering algorithms suffer nevertheless from three main computational bottlenecks: the construction of the similarity matrix $\boldsymbol{W}$; the partial eigenvalue decomposition of the corresponding graph Laplacian matrix $\boldsymbol{\mathcal{L}}$; and k-means or other discretization procedures.

In this paper, inspired by recent advances in the field of anchor graph construction [28]–[32] and nonnegative relaxation [33], [34], we present a scalable graph-based clustering with nonnegative relaxation (SGCNR) to circumventing the three computational bottlenecks in large HSI data sets.

The main contributions of our work are as follows.

(i) We combine orthonormal constraint with nonnegative relaxation to build a novel graph-based clustering model. In this case, a simple and efficient algorithm is proposed to solve our objective function by using the augmented Lagrangian multiplier (ALM) method.

(ii) We present the use of the anchor graph model for speeding up the construction of the similarity matrix $\boldsymbol{W}$ and subsequent optimization. The overall complexity of this model reduces to $O(nd \log m + nK^2 + nKc + K^3)$, where m and c are the numbers of anchors and nearest neighbors. In particular, for large HSIs, $n \gg m$, $n \gg d$, and $n \gg K$, this model greatly reduces the computational complexity. In addition, the similarity matrix is optimized as variable, and thus, we obtain a more reliable similarity matrix which can improve the final clustering performance.

(iii) By nonnegative relaxation, we can directly obtain the clustering indicators, without resort to k-means or other discretization procedures as traditional graph-based clustering algorithms have to do.

The rest of this paper is organized as follows. In Section II, we briefly introduce the prior work on the graph-based clustering approaches. Our SGCNR model is proposed in Section III. In Section IV, we validate our SGCNR algorithm and make comparisons with other approaches on different HSI data sets. We finally conclude this paper in Section V.

II. Brief Review of the Prior Work

Graph-based clustering has been thoroughly studied and many different algorithms have been investigated [35]. In this section, we first introduce two kinds of representative graph-based clustering algorithms: Ratio Cut [24] and Normalized Cut [25]. Then, a detailed analysis of their limitations is given.

A. Ratio Cut and Normalized Cut

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^T \in \mathbb{R}^{n \times d}$ denote data matrix, where n is the number of data points and d is the dimension of features. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ be an undirected weighted graph with vertex set $\boldsymbol{X}$. $\mathcal{V}$ denotes the set of n vertexes and each data point $\boldsymbol{x}_i$ is represented as a vertex on the affinity graph $\mathcal{G}$. $\mathcal{E}$ denotes the set of edges and each edge represents the similarity relationship of one pair of vertexes. $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ denotes the similarity matrix such that $W_{ij} = W_{ji} \geq 0$ is the weight of the edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The graph's Laplacian matrix $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ is denoted as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where the degree matrix $\boldsymbol{D}$ is defined as a diagonal matrix with the ith diagonal element as $d_i = \sum_j W_{ij}$. The graph's normalized Laplacian matrix is represented as $\tilde{\boldsymbol{L}} = \boldsymbol{I} - \boldsymbol{D}^{-1/2} \boldsymbol{W} \boldsymbol{D}^{-1/2}$, where $\boldsymbol{I}$ is an identity matrix.

Ratio Cut [24] and Normalized Cut [25] minimize the objective functions, respectively,

$$J = \sum_{1 \leq p < q \leq K} \frac{s(C_p, C_q)}{\rho(C_p)} + \frac{s(C_p, C_q)}{\rho(C_q)} = \sum_{k=1}^{K} \frac{s(C_k, \bar{C}_k)}{\rho(C_k)}, \tag{1}$$

$$\rho(C_k) = \begin{cases} |C_k| & \text{for Ratio Cut} \\ \sum_{i \in C_k} d_i & \text{for Normalized Cut} \end{cases} \tag{2}$$

where K is the number of clusters, C_k is the set of kth cluster, $\bar{C}_k$ is the complement of subset C_k in the graph $\mathcal{G}$, and $s(A, B) = \sum_{i \in A} \sum_{j \in B} W_{ij}$, $|C_k|$ is the number of elements in the set C_k and $d_i = \sum_j W_{ij}$.

We now reformulate the objective functions for these two different graph-based clustering algorithms. Let $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_K] \in \mathbb{R}^{n \times K}$ be the cluster indicator matrix where the ith element of $\boldsymbol{q}_k$ is 1 if the data point $\boldsymbol{x}_i$ belongs to cluster k, and 0 otherwise. We can easily see that $s(C_k, \bar{C}_k) = \sum_{i \in C_k} \sum_{j \in \bar{C}_k} W_{ij} = \boldsymbol{q}_k^T(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{q}_k$, $|C_k| = \boldsymbol{q}_k^T \boldsymbol{q}_k$ and $\sum_{i \in C_k} d_i = \boldsymbol{q}_k^T \boldsymbol{D} \boldsymbol{q}_k$. The objective functions of Ratio Cut and Normalized Cut can be rewritten as

$$J_{\text{rcut}} = \sum_{k=1}^{K} \frac{\boldsymbol{q}_k^T(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{q}_k}{\boldsymbol{q}_k^T \boldsymbol{q}_k}, \tag{3}$$

$$J_{\text{ncut}} = \sum_{k=1}^{K} \frac{\boldsymbol{q}_k^T(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{q}_k}{\boldsymbol{q}_k^T \boldsymbol{D} \boldsymbol{q}_k}. \tag{4}$$

We can clearly see the connections and differences between J_{rcut} and J_{ncut}. Let $\boldsymbol{F} = [\boldsymbol{q}_1/\|\boldsymbol{q}_1\|_2, \ldots, \boldsymbol{q}_K/\|\boldsymbol{q}_K\|_2]$ for Ratio Cut and $\boldsymbol{F} = [\boldsymbol{D}^{1/2}\boldsymbol{q}_1/\|\boldsymbol{D}^{1/2}\boldsymbol{q}_1\|_2, \ldots, \boldsymbol{D}^{1/2}\boldsymbol{q}_K/\|\boldsymbol{D}^{1/2}\boldsymbol{q}_K\|_2]$ for Normalized Cut, respectively. It is easy to verify that $\boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}$. Note that the objective functions J_{rcut} and J_{ncut} share the same formulation

$$\min_{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}} \operatorname{Tr}(\boldsymbol{F}^T \boldsymbol{\mathcal{L}} \boldsymbol{F}) \tag{5}$$

where $\boldsymbol{\mathcal{L}}$ is the graph's Laplacian matrix $\boldsymbol{L}$ in Ratio Cut [24] and is the graph's normalized Laplacian matrix $\tilde{\boldsymbol{L}}$ in Normalized Cut [25], respectively.

Now let us first look at how to minimize problem (5) with respect to $\boldsymbol{F}$. Note that the elements of $\boldsymbol{F}$ are constrained to be discrete values, which makes this problem hard to solve. A well-known solution to this problem is to relax the matrix $\boldsymbol{F}$ from the discrete values to continuous ones, meanwhile preserves the orthonormal constraints $\boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}$. The optimal solution $\boldsymbol{F}$ of (5) is formed by the K eigenvectors of $\boldsymbol{\mathcal{L}}$ corresponding to the K smallest eigenvalues. Since $\boldsymbol{F}$ is now in relaxed continuous form and has mixed signs, we have to resort to other discretization procedures, such as k-means, to obtain the final clustering results.

Note that not only Ratio Cut and Normalized Cut but also many existing graph-based clustering algorithms follow the same scheme which has three key parts [36]. First, compute weights W_{ij} that models the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and give rise to the similarity matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$. Second, compute the K eigenvectors $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_K] \in \mathbb{R}^{n \times K}$ of the Laplacian matrix $\boldsymbol{\mathcal{L}}$ associated with $\mathcal{G}$. Finally, run k-means using the rows of $\boldsymbol{F}$ as feature vectors to partition the n data points into K clusters.

B. Limitations of Ratio Cut and Normalized Cut

Ratio Cut and Normalized Cut are two representative algorithms in the family of graph-based clustering methods. These two algorithms have shown state-of-the-art clustering performance and been widely applied to many applications. However, they have limitations in the large HSIs clustering problem, which are analyzed below.

The first limitation comes from the underlying computational complexity which is of great importance for the clustering of large HSIs. The complexity mainly arises from two aspects. The first is the similarity matrix construction which takes $O(n^2 d)$ time complexity, and the second is the eigenvalue decomposition of the (normalized) Laplacian matrix which is implemented in $O(n^2 K)$, where n, d, and K are the number of samples, features, and clusters, respectively. Both of them are an unbearable burden for large HSIs clustering problem.

The second limitation is that the matrix $\boldsymbol{F}$ should be relaxed from the discrete values to continuous ones which make problem (5) solvable. However, the obtained solution $\boldsymbol{F}$ has mixed signs which could severely deviate from the true solution and have to resort to other clustering methods, such as k-means, to obtain final clustering results.

III. Scalable Graph-Based Clustering With Nonnegative Relaxation

To address the above issues in Ratio Cut and Normalized Cut, we accordingly propose two improvements. First, we introduce an anchor set and graph learning strategy into the graph construction, which not only speed up the graph construction and subsequent optimization but also result in the reliable affinity graph and improve the final clustering performance. Second, we add nonnegative relaxation on the matrix $\boldsymbol{F}$ which is directly used to extract the clustering indicators, without resort to other discretization procedures. In addition, a simple and efficient algorithm is proposed to solve this new problem with the nonnegative constraint rigorously. More details are given in the following.

A. Anchor Graph Construction

Recent studies [28], [29], [37], [38] adopt anchor-based strategy to construct similarity matrix $\boldsymbol{W}$. There are two key parts in anchor-based strategy. The first part is the generation of anchors, in which $m(m \ll n)$ anchors needed to be generated from n data points. In general, it can be achieved by random selection or using k-means method. Random selection selects m anchors by random sampling from data points with computational complexity $O(1)$. Although random selection cannot guarantee that the selected m anchors are always good, it is extremely fast for large HSI clustering. k-means method makes use of m clustering centers as anchors. The clustering centers are more representative anchors, but the computational complexity of k-means is $O(ndmt)$, where t is the number of iterations, which makes it impossible to large HSI clustering.

The second part is the design of a matrix $\boldsymbol{Z} \in \mathbb{R}^{n \times m}$ that measures the similarity between data points and anchors. Let $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]^T \in \mathbb{R}^{m \times d}$ denotes the generated m anchors, Liu *et al.* [28] define the (i, j)th element of $\boldsymbol{Z}$ based on a kernel function $K_\sigma(\cdot)$ with a bandwidth σ

$$z_{ij} = \frac{K_\sigma(\boldsymbol{x}_i, \boldsymbol{u}_j)}{\sum_{s \in \Phi_i} K_\sigma(\boldsymbol{x}_i, \boldsymbol{u}_s)}, \quad \forall j \in \Phi_i, \tag{6}$$

where $\Phi_i \subset \{1, \ldots, m\}$ denotes the set saving the indexes of the k nearest anchors of $\boldsymbol{x}_i$, Gaussian kernel

$K_\sigma(\boldsymbol{x}_i, \boldsymbol{u}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{u}_j\|_2^2/2\sigma^2)$ is adopted in [28], but kernel-based methods always bring extra parameters, e.g., bandwidth σ.

Intuitively, if $\boldsymbol{Z}$ is learned as variable, the performance of subsequent learning task will be improved. Therefore, in this work, the matrix $\boldsymbol{Z}$ is defined as a variable and can be optimized by solving the following problem:

$$\min_{\boldsymbol{Z1}=\boldsymbol{1}, z_{ij}\geq 0} \sum_{i=1}^{n}\sum_{j=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{u}_j\|_2^2 z_{ij} + \gamma \,\|\boldsymbol{Z}\|_F^2 \tag{7}$$

where γ is the regularization parameter and $\boldsymbol{1}$ denotes a column vector with all elements as one. Problem (7) can be solved separately for each z_i as follows:

$$\min_{z_i^T \boldsymbol{1}=1, z_{ij}\geq 0} \sum_{j=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{u}_j\|_2^2 z_{ij} + \gamma\, z_{ij}^2 \tag{8}$$

where z_i^T denotes the ith row of $\boldsymbol{Z}$. Denote $d_{ij}^x = \|\boldsymbol{x}_i - \boldsymbol{u}_j\|_2^2$ and $\boldsymbol{d}_i \in \mathbb{R}^{m\times 1}$ is a vector with the jth element as d_{ij}^x, then problem (8) can be written in vector form as

$$\min_{z_i} \frac{1}{2}\left\| z_i + \frac{\boldsymbol{d}_i}{2\gamma}\right\|_2^2 \quad \text{s.t.} \quad z_i^T \boldsymbol{1} = 1, z_{ij} \geq 0. \tag{9}$$

The Lagrangian function of problem (9) is

$$\mathcal{L}(z_i, \eta, \boldsymbol{\beta}) = \frac{1}{2}\|z_i + \frac{\boldsymbol{d}_i}{2\gamma}\|_2^2 - \eta\big(z_i^T \boldsymbol{1} - 1\big) - \boldsymbol{\beta}^T z_i \tag{10}$$

where η is a scalar and $\boldsymbol{\beta}$ is a vector, both of which are Lagrangian multipliers that to be determined. It is preferred to learn a sparse z_i which has exactly c nonzero values. Without loss of generality, suppose $d_{i1}, d_{i1}, \ldots, d_{im}$ are ordered from small to large. According to the Karush–Kuhn–Tucker condition, the parameter γ can be set as $\gamma = \frac{c}{2}d_{i,c+1} - \frac{1}{2}\sum_{j=1}^{c} d_{ij}$ and the solution to (9) is

$$z_{ij} = \frac{d_{i,c+1} - d_{ij}}{c d_{i,c+1} - \sum_{j=1}^{c} d_{ij}} \tag{11}$$

which have the following advantages.

1) The learned z_{ij} is scale invariant. If the data points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ are scaled by an arbitrary scalar α, and anchors $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\}$ is scaled accordingly. For example, let $\boldsymbol{x}_i$ be $\alpha \cdot \boldsymbol{x}_i$ for each i, then $\boldsymbol{u}_j$ and d_{ij} are changed to be $\alpha \cdot \boldsymbol{u}_j$ and $\alpha \cdot d_{ij}$ for each i, j, but the z_{ij} computed by (11) will not be changed. While in Gaussian function, the z_{ij} will be changed in this case, which makes the bandwidth σ difficult to tune.
2) From (11), we see that the parameter c is much easier to tune than γ since c is an integer and has explicit meaning. In most cases, $c < 10$ is likely to yield reasonable results. This property is important since the tuning of hyperparameters remains a difficult and open problem in many learning tasks.
3) The learned $\boldsymbol{Z}$ is naturally c-sparse and the computation burden of subsequent optimization can be alleviated largely.

Thus, we only need $O(ndm)$ to compute the matrix $\boldsymbol{Z}$. To have a clear impression, we take an illustrative example of constructing $\boldsymbol{Z}$ in Fig. 1. The construction is built on a three-ring synthetic data, which consist of 5000 data points. We adopt k-means method to choose 200 anchors from the original data points.

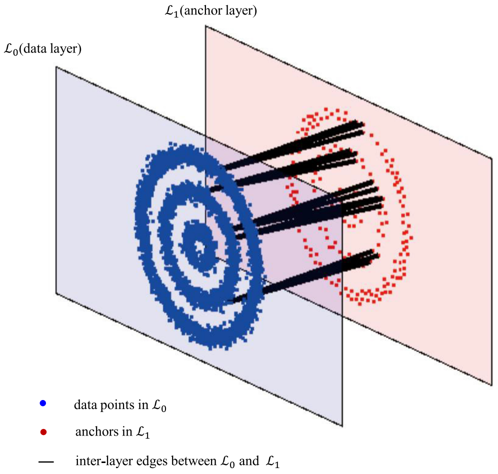

$\mathcal{L}_1$(anchor layer)
$\mathcal{L}_0$(data layer)
data points in $\mathcal{L}_0$
anchors in $\mathcal{L}_1$
inter-layer edges between $\mathcal{L}_0$ and $\mathcal{L}_1$

Fig. 1. Construction of $\boldsymbol{Z}$. Blue points: original data points. Red points: anchors. For convenience, only a tiny fraction of interlayer edges, which presents the weights between the original data points and anchors, are shown.

As we obtain the matrix $\boldsymbol{Z}$, the similarity matrix $\boldsymbol{W}$ then can be obtained by [28]

$$\boldsymbol{W} = \boldsymbol{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^T, \tag{12}$$

with the i-th diagonal element in which the diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ is defined as $\boldsymbol{\Lambda}_{jj} = \Sigma_{i=1}^{n} z_{ij}$. From (12), the (i, j)-th element of $\boldsymbol{W}$ is $W_{ij} = z_i^T \boldsymbol{\Lambda}^{-1} z_j$ which satisfies $W_{ij} = W_{ji}$, thus $\boldsymbol{W}$ is a symmetric matrix. Note that $z_{ij} \geq 0$ and $z_i^T \boldsymbol{1} = 1$, it can be verified that $\boldsymbol{W}$ is a double stochastic matrix [39] which satisfies $W_{ij} \geq 0$ and

$$\sum_{j=1}^{n} W_{ij} = \sum_{j=1}^{n} z_i^T \boldsymbol{\Lambda}^{-1} z_j = z_i^T \sum_{j=1}^{n} \boldsymbol{\Lambda}^{-1} z_j = z_i^T \boldsymbol{1} = 1. \tag{13}$$

Thus, the similarity matrix $\boldsymbol{W}$ is automatically normalized and the degree matrix $\boldsymbol{D} = \boldsymbol{I}$.

B. Scalable Graph-Based Clustering With Nonnegative Relaxation

To solve problem (5), the traditional approach relaxes the matrix $\boldsymbol{F}$ from the discrete values to continuous ones and makes it satisfies orthonormal constraint $\boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}$. To avoid the second limitation, we need to consider more accurate relaxation. Note that the matrix $\boldsymbol{F}$ is a nonnegative matrix, a more accurate relaxation is adding the nonnegative constraints on the matrix $\boldsymbol{F}$

$$\boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}, \boldsymbol{F} \geq 0. \tag{14}$$

Theorem 1: If the matrix $\boldsymbol{F}$ satisfies the orthonormal constraint $\boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}$ and nonnegative constraint $\boldsymbol{F} \geq 0$ simultaneously. For each row of $\boldsymbol{F}$, only one element is positive and others are zeros, thus $\boldsymbol{F}$ is very close to the ideal cluster indicator matrix.

Algorithm 1 Algorithm to Solve Problem (17)

```
Set $1 < \rho < 2$. Initialize $\mu > 0$, $\boldsymbol{\Lambda}$
while not converge do
    1) Update $\boldsymbol{X}$ by $\min_{\boldsymbol{X}} f(\boldsymbol{X}) + \frac{\mu}{2}\|h(\boldsymbol{X}) + \frac{1}{\mu}\boldsymbol{\Lambda}\|_F^2$.
    2) Update $\boldsymbol{\Lambda}$ by $\boldsymbol{\Lambda} = \boldsymbol{\Lambda} + \mu h(\boldsymbol{X})$.
    3) Update $\mu$ by $\mu = \rho\mu$.
end while
```

Proof: Let $\boldsymbol{f}_i$ be the ith column of the matrix $\boldsymbol{F}$ and $\boldsymbol{f}_j$ $(j \neq i)$ denotes any column of the $\boldsymbol{F}$. From orthonormal constraint $\boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}$, we know that

$$\boldsymbol{f}_i^T\boldsymbol{f}_j = \sum_{r=1}^{n} f_{ri} f_{rj} = 0. \tag{15}$$

For $\boldsymbol{F} \geq 0$, each element of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ is nonnegative. Thus, $f_{ri}f_{rj} = 0$ for each r. Suppose the rth element of $\boldsymbol{f}_i$ is positive. The corresponding rth element of $\boldsymbol{f}_j$ must be 0. ■

From Theorem 1, we conclude that if $\boldsymbol{F}$ satisfies the orthonormal and nonnegative constraints simultaneously, the obtained $\boldsymbol{F}$ can be used directly to assign cluster labels to data points. Therefore, we combine orthonormal constraint with nonnegative relaxation to build a model called SGCNR as follows:

$$\min_{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}, \boldsymbol{F}\geq 0} \mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{F}). \tag{16}$$

According to (12), $\boldsymbol{W}$ can be written as $\boldsymbol{W} = \boldsymbol{B}\boldsymbol{B}^T$, where $\boldsymbol{B} = \boldsymbol{Z}\boldsymbol{\Lambda}^{-1/2}$. In addition, $\boldsymbol{W}$ is automatically normalized, thus $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T$. We use the ALM method [40]–[42] to solve this problem.

1) Brief Description of ALM Method: Consider the constrained optimization problem

$$\min_{h(\boldsymbol{X})=\boldsymbol{0}} f(\boldsymbol{X}). \tag{17}$$

The algorithm using the ALM method to solve problem (17) is described in Algorithm 1. It has been proven that under some rather general conditions, Algorithm 1 converges Q-linearly to the optimal solution [42]. This property makes the ALM method very attractive.

2) Solving Problem (16) Using ALM Method: Problem (16) is equivalently rewritten as

$$\min_{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}, \boldsymbol{F}=\boldsymbol{G}, \boldsymbol{G}\geq 0} \mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{G}). \tag{18}$$

According to step 1 in Algorithm 1, we need to solve the following problem:

$$\min_{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}, \boldsymbol{G}\geq 0} \mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{G}) + \frac{\mu}{2}\|\boldsymbol{F} - \boldsymbol{G} + \frac{1}{\mu}\boldsymbol{\Lambda}\|_F^2. \tag{19}$$

We can solve problem (19) by means of the alternative optimization method.

The first step is fixing $\boldsymbol{G}$ and solving $\boldsymbol{F}$. Then, problem (19) becomes

$$\min_{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}} \mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{G}) + \frac{\mu}{2}\|\boldsymbol{F} - \boldsymbol{G} + \frac{1}{\mu}\boldsymbol{\Lambda}\|_F^2. \tag{20}$$

Note that

$$\begin{aligned}
&\mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{G}) + \frac{\mu}{2}\|\boldsymbol{F} - \boldsymbol{G} + \frac{1}{\mu}\boldsymbol{\Lambda}\|_F^2 \\
&\Leftrightarrow \mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{G}) + \frac{\mu}{2}\mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{F}) - \frac{\mu}{2}\mathrm{Tr}\left(\boldsymbol{F}^T\left(\boldsymbol{G} - \frac{1}{\mu}\boldsymbol{\Lambda}\right)\right) \\
&\quad - \frac{\mu}{2}\mathrm{Tr}\left(\left(\boldsymbol{G} - \frac{1}{\mu}\boldsymbol{\Lambda}\right)^T\boldsymbol{F}\right) \\
&\quad + \frac{\mu}{2}\mathrm{Tr}\left(\left(\boldsymbol{G} - \frac{1}{\mu}\boldsymbol{\Lambda}\right)^T\left(\boldsymbol{G} - \frac{1}{\mu}\boldsymbol{\Lambda}\right)\right),
\end{aligned}$$

problem (20) is simplified to the following problem:

$$\min_{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}} \mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{M}) \tag{21}$$

where $\boldsymbol{M} = (1-\mu)\boldsymbol{G} - \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{G} + \boldsymbol{\Lambda}$. Suppose the singular value decomposition (SVD) of $\boldsymbol{M}$ is $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T$, where $\boldsymbol{U} \in \mathbb{R}^{n\times n}$, $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times K}$ and $\boldsymbol{V} \in \mathbb{R}^{K\times K}$, then we have

$$\begin{aligned}
\mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{M}) &= \mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T) \\
&= \mathrm{Tr}(\boldsymbol{\Lambda}\boldsymbol{V}^T\boldsymbol{F}^T\boldsymbol{U}) \\
&= \mathrm{Tr}(\boldsymbol{\Lambda}\boldsymbol{\Phi}) = \sum_i \lambda_{ii}\Phi_{ii}
\end{aligned} \tag{22}$$

where $\boldsymbol{\Phi} = \boldsymbol{V}^T\boldsymbol{F}^T\boldsymbol{U} \in \mathbb{R}^{K\times n}$, λ_{ii} and Φ_{ii} are the (i, i)th element of matrix $\boldsymbol{\Lambda}$ and $\boldsymbol{\Phi}$ respectively.

Note that $\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{I}_K$, where $\boldsymbol{I}_K$ is an K by K identity matrix, so $-1 \leq \Phi_{ii} \leq 1$. On the other hand, $\lambda_{ii} \geq 0$ since λ_{ii} is singular value of $\boldsymbol{M}$. Therefore, $\mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{M}) = \sum_i \lambda_{ii}\Phi_{ii} \geq -\sum_i \lambda_{ii}$, and when $\Phi_{ii} = -1 (1 \leq i \leq K)$, the equality holds. That is to say, $\mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{M})$ reaches the minimum when $\boldsymbol{\Phi} = [-\boldsymbol{I}_K, \boldsymbol{0}]$. Recall that $\boldsymbol{\Phi} = \boldsymbol{V}^T\boldsymbol{F}^T\boldsymbol{U}$, thus the optimal solution to problem (21) is

$$\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Phi}^T\boldsymbol{V}^T = \boldsymbol{U}[-\boldsymbol{I}_K, \boldsymbol{0}]\boldsymbol{V}^T. \tag{23}$$

The second step is fixing $\boldsymbol{F}$ and solving $\boldsymbol{G}$. Then, problem (19) becomes

$$\min_{\boldsymbol{G}\geq 0} \mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{G}) + \frac{\mu}{2}\|\boldsymbol{Y} - \boldsymbol{G}\|_F^2, \tag{24}$$

where $\boldsymbol{Y} = \boldsymbol{F} + (1/\mu)\boldsymbol{\Lambda}$. Note that

$$\begin{aligned}
&\mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{G}) + \frac{\mu}{2}\|\boldsymbol{Y} - \boldsymbol{G}\|_F^2 \\
&= \frac{1}{2}\mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{G}) + \frac{1}{2}\mathrm{Tr}(\boldsymbol{G}^T\boldsymbol{L}\boldsymbol{F}) + \frac{\mu}{2}\mathrm{Tr}(\boldsymbol{Y}^T\boldsymbol{Y}) \\
&\quad - \frac{\mu}{2}\mathrm{Tr}(\boldsymbol{Y}^T\boldsymbol{G}) - \frac{\mu}{2}\mathrm{Tr}(\boldsymbol{G}^T\boldsymbol{Y}) + \frac{\mu}{2}\mathrm{Tr}(\boldsymbol{G}^T\boldsymbol{G}) \\
&= \frac{\mu}{2}\mathrm{Tr}(\boldsymbol{G}^T\boldsymbol{G}) - \frac{\mu}{2}\mathrm{Tr}\left(\boldsymbol{G}^T\left(\boldsymbol{Y} - \frac{1}{\mu}\boldsymbol{L}\boldsymbol{F}\right)\right) \\
&\quad - \frac{\mu}{2}\mathrm{Tr}\left(\left(\boldsymbol{Y} - \frac{1}{\mu}\boldsymbol{L}\boldsymbol{F}\right)^T\boldsymbol{G}\right) + \frac{\mu}{2}\mathrm{Tr}(\boldsymbol{Y}^T\boldsymbol{Y}),
\end{aligned}$$

problem (24) can be written in the following form:

$$\min_{\boldsymbol{G}\geq \boldsymbol{0}} \|\boldsymbol{G} - \boldsymbol{H}\|_F^2, \tag{25}$$

where $\boldsymbol{H} = \boldsymbol{Y} - (1/\mu)\boldsymbol{L}\boldsymbol{F} = (1-(1/\mu))\boldsymbol{F} + \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{F} + (1/\mu)\boldsymbol{\Lambda}$. Note that the above problem is independent between different

Algorithm 2 Algorithm of SGCNR

```
Set $1 < \rho < 2$. Initialize $\mu > 0$, $\mathbf{\Lambda}$
while not converge do
    1) Update $\boldsymbol{F}$ by (23).
    2) Update $\boldsymbol{G}$ by the solution of problem (26).
    3) Update $\mathbf{\Lambda}$ by $\mathbf{\Lambda} = \mathbf{\Lambda} + \mu(\boldsymbol{F} - \boldsymbol{G})$.
    4) Update $\mu$ by $\mu = \rho\mu$.
end while
```

element G_{ij}, so we can solve the following problem separately for each element G_{ij}:

$$\min_{G_{ij} \geq 0} (G_{ij} - H_{ij})^2. \tag{26}$$

If $H_{ij} \geq 0$, the optimal solution of G_{ij} is equal to H_{ij}. If $H_{ij} < 0$, the optimal solution of G_{ij} is equal to 0.

As mentioned before, the solution $\boldsymbol{G}$ is very close to the ideal class indicator matrix and can be directly used to assign cluster labels to data points. Specifically, the ith data point $\boldsymbol{x}_i$ is assigned to cluster label l_i as $l_k = \max_k G_{ik}$.

Based on the ALM method in Algorithm 1, the detailed of our SGCNR algorithm is described in Algorithm 2.

C. Computational Complexity Analysis

We now analyze the computational complexity of SGCNR. Suppose we have a data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and m anchors are generated from $\boldsymbol{X}$. Thus, the main steps of SGCNR and the corresponding computational complexity are summarized as follows.

1) We need $O(1)$ to obtain m anchors by random selection.
2) In general, we need $O(ndm)$ to obtain the matrix $\boldsymbol{Z}$. This process can also be accelerated by the approximate nearest neighbor search technique and be efficiently implemented in $O(nd \log m)$ [31], [43].
3) Note that the learned $\boldsymbol{Z}$ is naturally c-sparse, thus the matrix $\boldsymbol{B}$ is also c-sparse. We need $O(nKc)$ to obtain the matrix $\boldsymbol{M}$. The SVD of the matrix $\boldsymbol{M}$ needs $O(nK^2 + K^3)$. Compute the matrix $\boldsymbol{F}$ by (23) with $O(nK^2)$. In sum, the computational complexity of this step is $O(nK^2)$.
4) We need $O(nKc)$ to obtain the matrix $\boldsymbol{H}$. Compute the matrix $\boldsymbol{G}$ by (26) with $O(nK)$. In sum, the computational complexity of this step is $O(nKc)$.

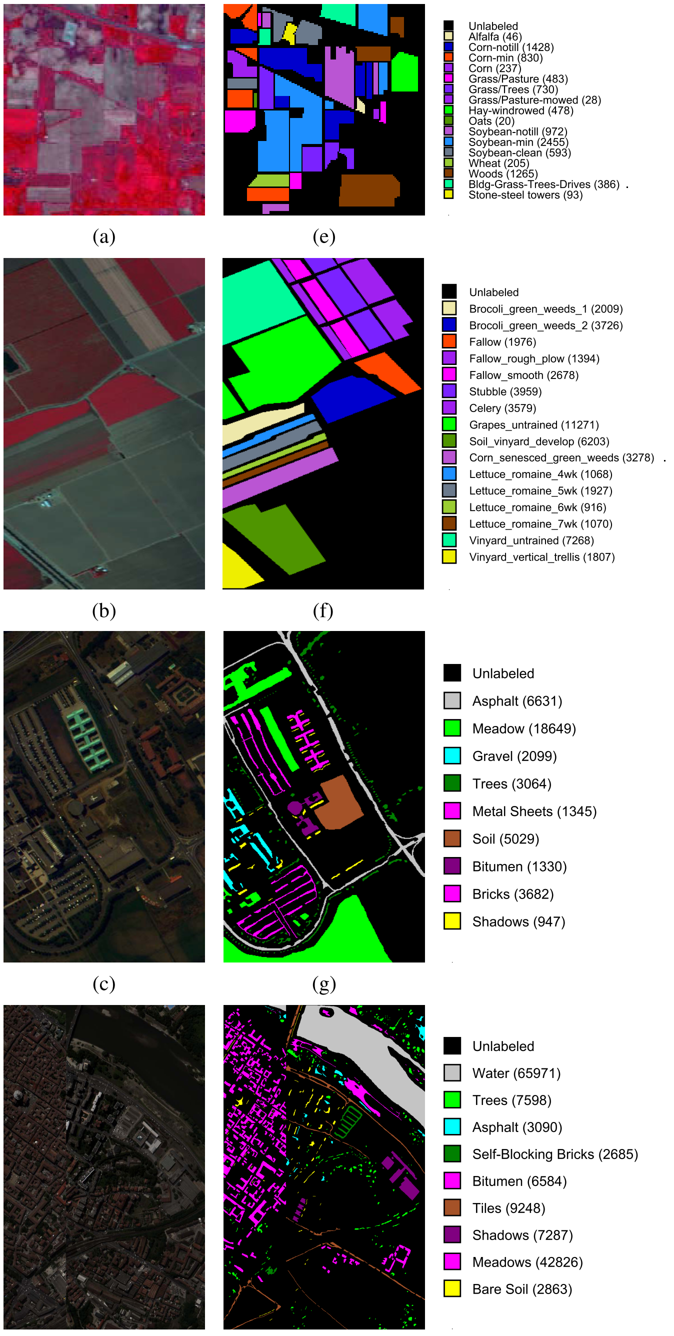
To sum up, the time complexity of SGCNR scales as $O(nd \log m + nK^2 + nKc + K^3)$, where d is the number of feature, m is the number of anchors, c is the number of nearest neighbors in adjacency estimation, and K is the number of classes.

IV. Experimental Results and Discussion

In this section, we investigate both the effectiveness and efficiency of our proposed SGCNR algorithm on the HSI data sets. The following clustering methods were selected as benchmarks: k-means [14], FCM [15], FCM_S1 [16], Ratio Cut [24] and Normalized Cut [25], and Non-negative matrix factorization (NMF) [33], [44]. All the experiments are implemented on a PC with E3-1505 v5 at 2.80 GHz and 32-GB RAM, MATLAB 2014a. Here, we use the following four HSI data sets with scales varying from 21 025 to 783 640. The descriptions of these data sets are given in the following [45], [46].

1) *Indian Pines:* The data set was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in 1992. The image scene contains 145 × 145 pixels and 220 spectral bands. A total of 20 water absorption and noisy bands (104–108, 150–163, and 220) were removed from the original 220 bands, leaving 200 spectral features for the experiment. The total number of samples is 21 025 and there are 16 classes in the data set. The false color image composition of bands 50, 27, and 17 and the ground truth map are provided in Fig. 1(a) and (e), respectively.
2) *Salinas:* The data set was also acquired by the AVIRIS sensor in 1998. The image scene contains 512 × 217 pixels and 224 spectral bands. A total of 20 water absorption bands (108–112, 154–167, and 224) were removed from the original 224 bands, leaving 204 spectral features for the experiment. The total number of samples is 111 104 and there are 16 classes in the data set. The false color image composition of bands 70, 27, and 17 and the ground truth map are shown in Fig. 1(b) and (f), respectively.
3) *Pavia University:* The data set was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor in 2001. The image scene contains 610 × 340 pixels and 115 spectral bands. Several spectral bands with noise are removed from the data set, leaving 103 bands to be used in the experiments. The total number of samples is 207 340 and there are 9 classes in the data set. The false color image composition of bands 60, 30, and 2 and the ground truth map are shown in Fig. 1(c) and (g), respectively.
4) *Pavia Center:* The data set was acquired by the ROSIS sensor. The image scene contains 1096×1096 pixels and 102 spectral bands. Since some of the samples in the image contain no information and have to be discarded from the original image, leaving 1096 × 715 pixels for the experiment. The total number of samples is 783 640 and there are 9 classes in the data set. The false color image composition of bands 40, 30, and 20 and the ground truth map are shown in Fig. 1(d) and (h), respectively.

The above data sets are categorized into small, medium, and large sizes. Specifically, in our experiments, we regard Indian Pines as a small-sized data set, Salinas as a medium-sized data set, Pavia University and Pavia Center as large-sized data sets. In addition, we set the number of clusters to be the ground truth in each data set. For k-means [14], FCM [15] and FCM_S1 [16], all the parameters of this method were manually tuned to the optimum. Meanwhile, we repeat these three methods 20 times and report the best result for each method. For those graph-based clustering methods calling for an input of a similarity matrix, like Ratio Cut, Normalized Cut, and NMF, the graph is constructed with the self-tune



(a) (e)

(b) (f)

(c) (g)

(d) (h)

Fig. 2. False color images and ground truth maps of four data sets. (a) Indian Pines image of bands 50, 27, and 17. (b) Salinas image of bands 70, 27, and 17. (c) Pavia University image of bands 60, 30, and 2. (d) Pavia Center image of bands 40, 30, and 20. (e)–(h) Corresponding ground truth maps. (Note that the number of each class is shown in brackets.)

Gaussian method [47] and the number of neighbors is set to 5. In order to thoroughly evaluate the clustering performance of each method, both visual clustering maps and quantitative evaluations (including user's accuracy, average accuracy (AA), overall accuracy (OA), and kappa coefficient) are given for each experiment.

A. Small-Size Data Set

We conduct experiments on the Indian Pines data set to validate the effectiveness of the proposed SGCNR algorithm. For the SGCNR algorithm, the number of anchors is set to $m = 500$ and the number of neighbors is set to $k = 5$. In addition, the parameters in the ALM method is empirically set to $\rho = 1.1$ and $\mu = 0.1$, respectively. The clustering maps obtained with each clustering method are shown from Fig. 3(a)–(g), and the corresponding quantitative evaluations of the clustering results are provided in Table I. In the table, the optimal value of each row is shown in bold, and the second best result is underlined.

From Fig. 3 and Table I, it can be clearly observed that the clustering result of NMF is very poor and contains a large number of misclassifications, especially for the three classes of soybean-min, corn-notill, and soybean-notill. Compared with NMF, Ratio Cut and Normalized Cut have higher precision, and the kappa coefficient is improved from 0.2367 to 0.2870 and 0.2873. We see that the three graph-based clustering methods Ratio Cut, Normalized Cut, and NMF obtain inferior clustering performances; however, in contrast k-means, FCM and FCM_S1 improve the clustering performance to obtain much smoother clustering maps. The proposed SGCNR method obtains the highest precision, with the best OA of 45.30% and a kappa coefficient of 0.3730. In particular, for the first four classes of soybean-min, corn-notill, woods, and soybean-notill, the SGCNR method obtains the highest or second highest precision of 69.33%, 43.98%, 48.77%, and 37.35%.

Table II shows the running time on the running time of each clustering method on the four HSI data sets. From this table, we see that the running time of k-means, FCM, FCM_S1, and SGCNR is of the same order of magnitude. Especially for the four graph-based clustering methods, the SGCNR only needs 7.3 s, which is faster than the Ratio Cut, Normalized Cut, and NMF. Note that the Ratio Cut, Normalized Cut, and NMF can work on the Indian Pines data set which does not belong to a large scale HSI data set because the total number of samples is only 21 025.

B. Medium-Size Data Set

Experiments are conducted on the Salinas data set. The parameters of the SGCNR algorithm needed to be set are the number of anchors $m = 500$ and the number of neighbors $k = 5$. In addition, the parameters in the ALM method is empirically set to $\rho = 1.1$ and $\mu = 0.1$, respectively.

From the clustering maps shown in Fig. 4 and the corresponding quantitative evaluations in Table III, it can be clearly observed that NMF obtain poor clustering results containing numerous misclassifications, with the lowest OA of 56.61% and kappa coefficient of 0.5048, respectively. Ratio Cut and Normalized Cut perform better by significantly decreasing the misclassifications to achieve the improvement of almost 7% and 10% in OA. It can also be seen that k-means, FCM, and FCM_S1 generate smoother clustering results than NMF in this scene, with the 11%, 8%, and 6% improvement in OA. The proposed SGCNR method obtains the best visual result and the highest accuracy by effectively distinguishing almost all classes, with the best OA of 70.08% and a kappa coefficient of 0.6671. It is worth noting that the Lettuce_4wk class has been effectively distinguished to some extent, while the recognition

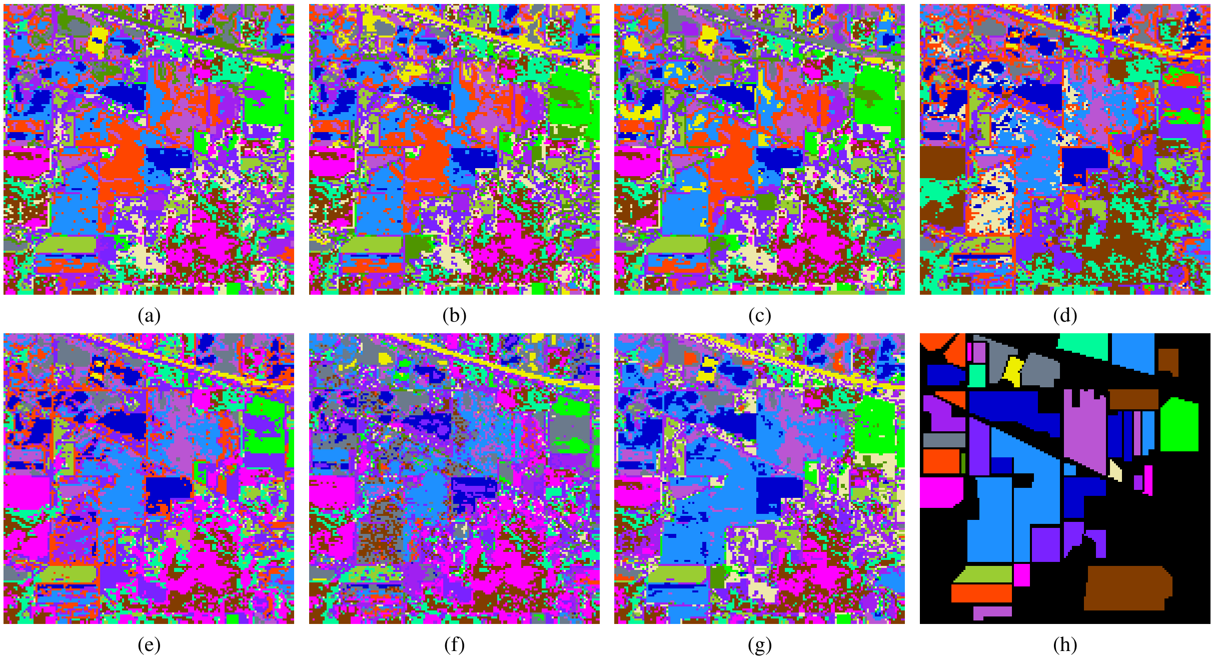

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 3. Clustering maps on the Indian Pines data set. (a) *k*-means. (b) FCM. (c) FCM_S1. (d) Rcut (e) Ncut. (f) NMF. (g) SGCNR. (h) Ground truth.

TABLE I
QUANTITATIVE EVALUATIONS ON THE INDIAN PINES HSI DATA SET

Method	Class	*k*-means	FCM	FCM_S1	Rcut	Ncut	NMF	SGCNR
User's accuracy (%)	Alfalfa	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Corn-notill	28.01	28.01	26.54	43.70	43.28	23.88	**43.98**
	Corn-min	40.24	40.60	39.04	**46.99**	46.87	15.18	1.33
	Corn	14.77	16.88	**21.10**	8.86	9.70	2.53	14.77
	Grass/Pasture	49.69	49.69	52.17	65.22	**65.42**	24.22	49.28
	Grass/Trees	40.82	40.82	**46.58**	27.95	28.49	20.96	21.23
	Grass/Pasture-mowed	**82.14**	**82.14**	0.00	0.00	0.00	0.00	0.00
	Hay-windrowed	85.98	75.52	82.85	62.97	65.69	82.43	**95.82**
	Oats	0.00	**45.00**	35.00	0.00	0.00	0.00	15.00
	Soybean-notill	27.88	27.98	27.78	27.47	27.47	19.86	**37.35**
	Soybean-min	33.65	33.81	33.73	24.48	24.44	29.04	**69.33**
	Soybean-clean	14.33	13.83	15.51	11.13	10.79	**25.46**	17.54
	Wheat	97.07	**98.05**	**98.05**	96.10	96.59	80.98	96.10
	Woods	41.82	41.42	43.64	37.15	36.68	**62.85**	48.77
	Bldg-Grass-Trees-Drives	17.88	17.88	18.65	**19.43**	**19.43**	**19.43**	15.80
	Stone-steel towers	**87.10**	2.15	83.87	21.51	19.35	15.05	68.82
AA (%)		**41.34**	38.36	39.03	30.81	30.89	26.37	38.14
OA (%)		37.09	35.98	37.48	34.65	34.68	31.65	**45.30**
Kappa		0.3030	0.2907	0.3112	0.2870	0.2873	0.2367	**0.3730**

TABLE II
RUNNING TIME ON FOUR HSI DATA SETS. ("OM" MEANS "OUT-OF-MEMORY ERROR")

Dataset	*k*-means	FCM	FCM_S1	Rcut	Ncut	NMF	SGCNR
Indian Pines	**3.0s**	4.3s	6.2s	42.9s	24.2s	209.5s	7.3s
Salinas	**14.1s**	24.5s	37.1s	1641.1s	1013.6s	5432.3s	40.8s
Pavia University	**16.2s**	23.5s	33.8s	OM	OM	OM	117.9s
Pavia Centre	**59.6s**	93.3s	132.2s	OM	OM	OM	475.7s

level of other methods is close to zero except for Normalized Cut. In addition, the Lettuce_5wk class is also distinguished well, and its accuracy is close to 100%, which is beyond the ability of other methods.

From Table II, we see that the running time of *k*-means, FCM, FCM_S1, and SGCNR is of the same order of magnitude. In particular, the SGCNR only takes 40.8 s, which is 40, 25, and 133 times faster than the Ratio Cut, Normalized Cut, and NMF, respectively. Since the Salinas data set does not belong to a large scale HSI data set, Ratio Cut, Normalized Cut, and NMF can still work on this data set.

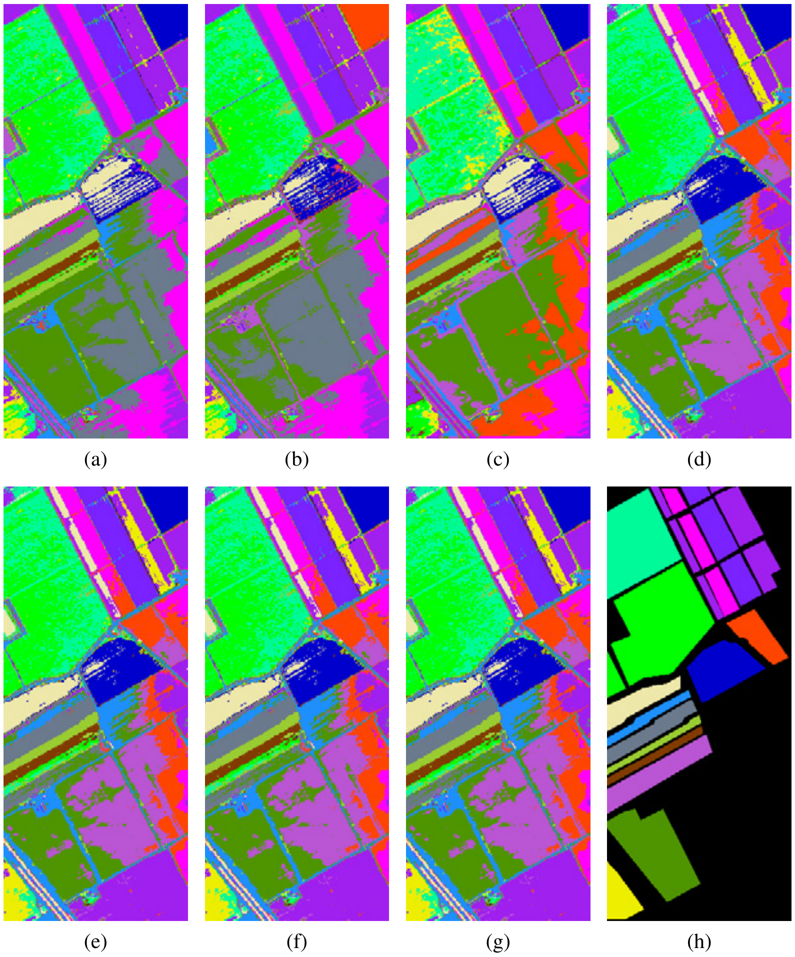


Fig. 4. Clustering maps on the Salinas data set. (a) k-means. (b) FCM. (c) FCM_S1. (d) Rcut (e) Ncut. (f) NMF. (g) SGCNR. (h) Ground truth.

TABLE III
QUANTITATIVE EVALUATIONS ON THE SALINAS DATA SET

Method	Class	k-means	FCM	FCM_S1	Rcut	Ncut	NMF	SGCNR
User's accuracy (%)	Brocoli_weeds_1	**99.75**	98.66	98.26	98.90	98.31	0.10	98.01
	Brocoli_weeds_2	39.59	67.85	43.67	85.64	43.28	**98.50**	92.03
	Fallow	0.00	0.00	79.71	0.00	0.00	**86.49**	62.25
	Fallow_rough	99.57	**99.71**	95.19	98.28	96.20	2.53	29.41
	Fallow_smooth	97.27	97.65	79.72	**98.28**	93.02	24.22	71.88
	Stubble	94.17	94.11	95.45	91.51	90.63	19.70	**97.90**
	Celery	97.04	96.90	**98.55**	93.10	88.74	98.24	65.88
	Grapes_untrained	68.25	67.40	31.33	**98.42**	97.87	82.43	64.15
	Soil	**97.16**	60.13	81.64	64.86	77.11	93.74	89.17
	Corn	0.00	0.88	27.39	56.44	**69.80**	0.00	2.23
	Lettuce_4wk	3.93	0.00	0.00	0.00	30.06	0.00	**65.26**
	Lettuce_5wk	92.01	72.29	51.89	0.00	0.00	0.00	**99.53**
	Lettuce_6wk	98.58	**98.69**	97.82	0.98	0.00	0.00	98.36
	Lettuce_7wk	88.88	88.79	87.57	0.00	0.00	0.00	**89.91**
	Vinyard_untrained	48.69	50.65	**72.95**	0.00	0.00	0.00	54.82
	Vinyard_trellis	47.04	46.82	15.61	61.93	73.93	15.99	**79.58**
AA (%)		67.00	65.03	66.05	53.02	57.30	43.74	**72.52**
OA (%)		67.34	64.36	62.54	63.24	66.54	56.61	**70.08**
Kappa		0.6357	0.6038	0.5885	0.5824	0.6213	0.5048	**0.6671**

C. Large-Size Data Sets

To demonstrate the scalability of the SGCNR algorithm, we conduct experiments on the Pavia University and Pavia Center data sets. The parameters of the SGCNR algorithm are set as $m = 1000$ and $k = 5$, respectively. In addition, the parameters in the ALM method is empirically set to $\rho = 1.1$ and $\mu = 0.1$, respectively. The clustering maps of the Pavia University and Pavia Center data sets are shown

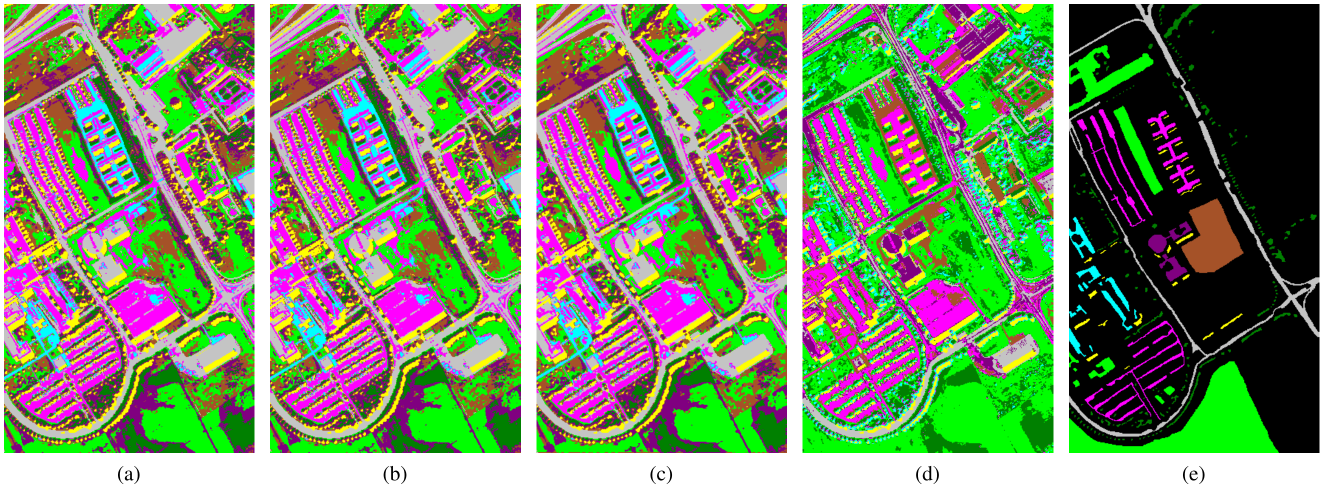


Fig. 5. Clustering maps on the Pavia University data set. (a) *k*-means. (b) FCM. (c) FCM_S1. (d) SGCNR. (e) Ground truth.

TABLE IV

QUANTITATIVE EVALUATIONS ON THE PAVIA UNIVERSITY DATA SET ("OM" MEANS "OUT-OF-MEMORY ERROR")

Method	Class	*k*-means	FCM	FCM_S1	Rcut	Ncut	NMF	SGCNR
User's accuracy (%)	Asphalt	$\underline{90.51}$	90.27	**91.77**	-	-	-	52.59
	Meadows	43.83	$\underline{43.93}$	41.98	-	-	-	**71.20**
	Gravel	0.10	0.10	0.19	-	-	-	0.00
	Trees	$\underline{63.67}$	62.21	61.49	-	-	-	**67.92**
	Painted metal sheets	48.25	49.22	$\underline{65.28}$	-	-	-	**99.48**
	Bare Soil	32.89	$\underline{33.25}$	**33.86**	-	-	-	18.77
	Bitumen	0.00	0.00	0.00	-	-	-	**91.05**
	Self-Blocking Bricks	94.24	94.38	$\underline{94.68}$	-	-	-	**96.20**
	Shadows	**100.00**	**100.00**	99.89	-	-	-	**100.00**
AA (%)		52.61	52.59	$\underline{54.35}$	-	-	-	**66.36**
OA (%)		$\underline{53.42}$	53.40	53.34	-	-	-	**62.72**
Kappa		0.4337	0.4333	$\underline{0.4347}$	OM	OM	OM	**0.5250**

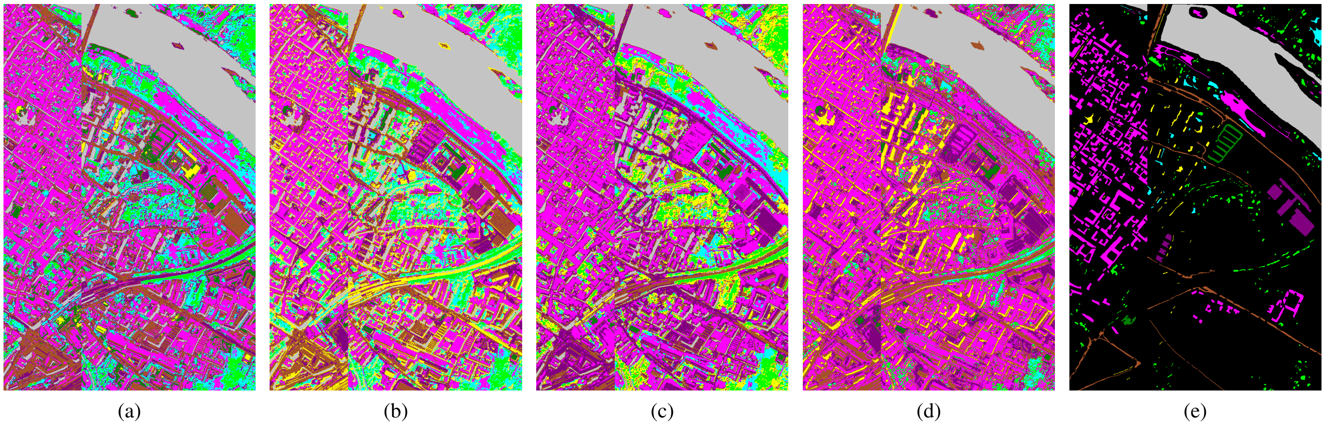


Fig. 6. Clustering maps on the Pavia Center data set. (a) *k*-means. (b) FCM. (c) FCM_S1. (d) SGCNR. (e) Ground truth.

in Figs. 5 and 6, respectively. The corresponding quantitative evaluations of the clustering results are provided in Tables IV and V, respectively.

As the growth of the scale of the Pavia University and Pavia Center data sets, the total number of samples increases to 207 340 and 783 640, respectively. By referring to Tables II, IV, and V, we see that the Ratio Cut, Normalized Cut, and NMF cannot work on the Pavia University and Pavia Center data sets due to "Out-of-Memory error." From Table II, we see that the running time of the SGCNR algorithm on the Pavia University and Pavia Center data sets is 117.9 and 475.7 s, respectively. Compared with *k*-means, FCM, and

TABLE V

QUANTITATIVE EVALUATIONS ON THE PAVIA CENTER DATA SET ("OM" MEANS "OUT-OF-MEMORY ERROR")

Method	Class	k-means	FCM	FCM_S1	Rcut	Ncut	NMF	SGCNR
User's accuracy (%)	Water	**99.32**	99.30	99.18	-	-	-	99.03
	Trees	65.06	63.89	**70.44**	-	-	-	53.00
	Asphalt	16.96	17.02	10.13	-	-	-	**40.58**
	Self-Blocking Bricks	**67.41**	11.51	12.33	-	-	-	11.40
	Bitumen	25.79	53.42	**56.11**	-	-	-	0.00
	Tiles	84.44	**84.97**	21.59	-	-	-	72.82
	Shadows	0.92	25.22	**82.71**	-	-	-	82.04
	Meadows	47.05	52.38	53.10	-	-	-	**96.82**
	Bare Soil	0.00	0.10	0.00	-	-	-	**99.79**
AA (%)		45.22	45.31	45.07	-	-	-	**61.72**
OA (%)		69.20	72.12	71.47	-	-	-	**86.36**
Kappa		0.5783	0.6166	0.6068	OM	OM	OM	**0.8049**

FCM_S1, the SGCNR algorithm runs a little longer, which is acceptable.

From Fig. 5 and Table IV, it can be clearly observed that k-means, FCM, and FCM_S1 obtain poor clustering results and contain significant amounts of misclassifications, with low OAs of 53.42%, 53.40%, and 53.34%, respectively. The proposed SGCNR algorithm achieves the best clustering result, both visually and quantitatively, with the best OA of 62.72% and kappa coefficient of 0.5250. It is pointed out that the meadows and Painted metal sheets classes are effectively distinguished, and the recognition level is much higher than that of other methods. In addition, the Bitumen class is also distinguished well and the accuracy is close to 91%, while the recognition level of other methods is zero.

From Fig. 6 and Table V, it can be seen that k-means, FCM, and FCM_S1 achieve inferior clustering results and contain a large number of misclassifications, with low kappas of 0.5783, 0.6166, and 0.6068, respectively. The proposed SGCNR algorithm achieves the best clustering result, both visually and quantitatively, with the best OA of 86.36% and kappa coefficient of 0.8049. Note that the meadows class is effectively distinguished, and the recognition level is much higher than that of other methods. In addition, the Bare Soil class is also distinguished well and the accuracy is close to 100%, while the recognition level of other methods is zero.

V. CONCLUSION

In this paper, we have presented a novel SGCNR algorithm to efficiently cluster the large HSI. The proposed SGCNR algorithm first constructs an anchor graph to accelerate the calculation process. The overall computational complexity is $O(nd\log m + nK^2 + nKc + K^3)$, which is a significant improvement compared with traditional graph-based clustering algorithms that need at least $O(n^2d + n^2K)$ or $O(n^2d + n^3)$. Then, the nonnegative relaxation term is added to directly extract the clustering indicators, without resort to k-means or other discretization procedures as traditional graph-based clustering algorithms have to do. Experimental results have demonstrated the efficiency and effectiveness of the proposed SGCNR algorithm for dealing with large HSI.

ACKNOWLEDGMENT

The authors would like to thank Prof. D. Landgrebe at Purdue University for providing the free downloads of the Indian Pines and Salinas AVIRIS images, and Prof. P. Gamba from the University of Pavia for providing the Pavia Center data set.

REFERENCES

[1] C.-I. Chang, *Hyperspectral Imaging Techniques for Spectral Detection and Classification*. New York, NY, USA: Springer, 2003.

[2] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithmsfor separable nonnegative matrix factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 698–714, Apr. 2014.

[3] H. Zhang, H. Zhai, L. Zhang, and P. Li, "Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3672–3684, Jun. 2016.

[4] R. Wang, F. Nie, and W. Yu, "Fast spectral clustering with anchor graph for large hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2003–2007, Nov. 2017.

[5] C. Deng, X. Liu, C. Li, and D. Tao, "Active multi-kernel domain adaptation for hyperspectral image classification," *Pattern Recognit.*, vol. 77, pp. 306–315, May 2018.

[6] K. Tasdemir and E. Merenyi, "A validity index for prototype-based clustering of data sets with complex cluster structures," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 4, pp. 1039–1053, Aug. 2011.

[7] Y. Zhong, A. Ma, and L. Zhang, "An adaptive memetic fuzzy clustering algorithm with spatial information for remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1235–1248, Apr. 2014.

[8] S. Mei, J. Hou, J. Chen, L.-P. Chau, and Q. Du, "Simultaneous spatial and spectral low-rank representation of hyperspectral images for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2872–2886, May 2018.

[9] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3893–3903, Aug. 2018.

[10] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[11] U. Maulik and I. Saha, "Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery," *Pattern Recognit.*, vol. 42, no. 9, pp. 2135–2149, Sep. 2009.

[12] H. Zhai, H. Zhang, L. Zhang, and P. Li, "Reweighted mass center based object-oriented sparse subspace clustering for hyperspectral images," *J. Appl. Remote Sens.*, vol. 10, no. 4, Nov. 2016, Art. no. 046014.

[13] H. Zhai, H. Zhang, X. Xu, L. Zhang, and P. Li, "Kernel sparse subspace clustering with a spatial max pooling operation for hyperspectral remote sensing data interpretation," *Remote Sens.*, vol. 9, no. 4, p. 335, 2017.

[14] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.

[15] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York, NY, USA: Plenum, 1981.

[16] S. Chen and D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1907–1916, Aug. 2004.

[17] S. Niazmardi, S. Homayouni, and A. Safari, "An improved FCM algorithm based on the SVDD for unsupervised hyperspectral data classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 831–839, Apr. 2013.

[18] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, Aug. 1996, pp. 226–231.

[19] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

[20] Y. Zhong, L. Zhang, and W. Gong, "Unsupervised remote sensing image classification using an artificial immune network," *Int. J. Remote Sens.*, vol. 32, no. 19, pp. 5461–5483, 2011.

[21] Y. Zhong, S. Zhang, and L. Zhang, "Automatic fuzzy clustering based on adaptive multi-objective differential evolution for remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 5, pp. 2290–2301, Oct. 2013.

[22] A. Ma, Y. Zhong, and L. Zhang, "Adaptive multiobjective memetic fuzzy clustering algorithm for remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4202–4217, Aug. 2015.

[23] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. NIPS*, 2002, pp. 849–856.

[24] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, "Spectral K-way ratio-cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 13, no. 9, pp. 1088–1096, Sep. 1994.

[25] 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.

[26] G. Chen and G. Lerman, "Spectral curvature clustering (SCC)," *Int. J. Comput. Vis.*, vol. 81, no. 3, pp. 317–330, Mar. 2009.

[27] H. Zhai, H. Zhang, L. Zhang, P. Li, and A. Plaza, "A new sparse subspace clustering algorithm for hyperspectral remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 43–47, Jan. 2017.

[28] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. ICML*, 2010, pp. 679–686.

[29] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, Aug. 2015.

[30] W. Zhu, F. Nie, and X. Li, "Fast spectral clustering with efficient large graph construction," in *Proc. ICASSP*, Mar. 2017, pp. 2492–2496.

[31] M. Wang, W. Fu, S. Hao, H. Liu, and X. Wu, "Learning on big graph: Label inference and regularization with anchor hierarchy," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 5, pp. 1101–1114, May 2017.

[32] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Proc. NIPS*, 2017, pp. 4129–4138.

[33] F. Nie, C. Ding, D. Luo, and H. Huang, "Improved minmax cut graph clustering with nonnegative relaxation," in *Machine Learning and Knowledge Discovery in Databases*, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds. Berlin, Germany: Springer, 2010, pp. 451–466.

[34] Y. Yang, H. T. Shen, F. Nie, R. Ji, and X. Zhou, "Nonnegative spectral clustering with discriminative regularization," in *Proc. AAAI*, Aug. 2011, pp. 555–560.

[35] M. C. V. Nascimento and A. C. P. L. F. de Carvalho, "Spectral methods for graph clustering—A survey," *Eur. J. Oper. Res.*, vol. 211, no. 2, pp. 221–231, 2011.

[36] X. Yang, C. Deng, X. Liu, and F. Nie, "New $\ell_{2,1}$-norm relaxation of multi-way graph cut for clustering," in *Proc. AAAI*, 2018, pp. 4374–4380.

[37] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proc. AAAI*, Aug. 2011, pp. 313–318.

[38] F. Nie, W. Zhu, and X. Li, "Unsupervised large graph embedding," in *Proc. AAAI*, Feb. 2017, pp. 2422–2428.

[39] R. Zass and A. Shashua, "Doubly stochastic normalization for spectral clustering," in *Proc. NIPS*, 2007, pp. 1569–1576.

[40] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," *Optimization*, vol. 5, no. 6, pp. 283–298, 1969.

[41] M. R. Hestenes, "Multiplier and gradient methods," *J. Optim. Theory Appl.*, vol. 4, no. 5, pp. 303–320, 1969.

[42] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, D. P. Bertsekas, Ed. New York, NY, USA: Academic, 1982.

[43] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.

[44] T. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *Proc. ICDM*, Dec. 2006, pp. 362–371.

[45] R. Wang, F. Nie, R. Hong, X. Chang, X. Yang, and W. Yu, "Fast and orthogonal locality preserving projections for dimensionality reduction," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 5019–5030, Oct. 2017.

[46] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1714–1754, Mar. 2018.

[47] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. NIPS*, 2005, pp. 1601–1608.

Rong Wang (M'12) 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 the Xi'an Research Institute of Hi-Tech, Xi'an, China, in 2004, 2007, and 2013, respectively.

From 2007 and 2013, he also studied for his Ph.D. degree in the Department of Automation, Tsinghua University, Beijing, China. He is currently an Associate Professor with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an. His research interests include machine learning and its applications.

Feiping Nie (M'17) 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 authored or coauthored more than 100 papers in the following journals and conferences: *International Journal of Computer Vision*, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, and ACM MM. His papers have been cited more than 10000 times and the H-index is 57. His research interests include machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval.

Dr. Nie is currently an Associate Editor or Program Committee Member for several prestigious journals and conferences.

Zhen Wang received the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2014.

From 2014 to 2016, he was a JSPS Senior Researcher with the Interdisciplinary Graduate School of Engineering Sciences, Kyushu University, Fukuoka, Japan. Since 2017, he has been a Full Professor with Northwestern Polytechnical University, Xi'an, China. He has authored or coauthored more than 100 scientific papers and obtained around 9000 citations. His research interests include network science, complex system, big data, evolutionary game theory, behavior decision, and behavior recognition.

Dr. Wang was a recipient of the National 1000 Talent Plan Program of China. He currently serves as an Editor or Academic Editor for seven journals.

Fang He received the B.S. and M.S. degrees from High-Tech Institute of Xi'an, Xi'an, China in 2014 and 2016, respectively, where she is currently pursuing the Ph.D. degree. Her research interests include machine learning and its applications.

Xuelong Li (M'02–SM'07–F'12) is a Full Professor with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China.