

Research Article

Exact k -Component Graph Learning for Image Clustering

Yufang Min^{1,2,3} and Yaonan Zhang^{1,3} 

¹Northwest Institute of Eco-Environment and Resources, CAS, Lanzhou 730000, Gansu, China

²University of Chinese Academy of Sciences, Beijing 100049, China

³Gansu Data Engineering and Technology Research Center for Resources and Environment, Lanzhou, Gansu 730000, China

Correspondence should be addressed to Yaonan Zhang; yaonan@lzb.ac.cn

Received 3 January 2020; Revised 14 May 2020; Accepted 15 May 2020; Published 6 August 2020

Academic Editor: Gen Q. Xu

Copyright © 2020 Yufang Min and Yaonan Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The performance of graph-based clustering methods highly depends on the quality of the data affinity graph as a good affinity graph can approximate well the pairwise similarity between data samples. To a large extent, existing graph-based clustering methods construct the affinity graph based on a fixed distance metric, which is often not an accurate representation of the underlying data structure. Also, they require postprocessing on the affinity graph to obtain clustering results. Thus, the results are sensitive to the particular graph construction methods. To address these two drawbacks, we propose a k -component graph clustering (k -GC) approach to learn an intrinsic affinity graph and to obtain clustering results simultaneously. Specifically, k -GC learns the data affinity graph by assigning the adaptive and optimal neighbors for each data point based on the local distances. Efficient iterative updating algorithms are derived for k -GC, along with proofs of convergence. Experiments on several benchmark datasets have demonstrated the effectiveness of k -GC.

1. Introduction

Clustering is one of the most fundamental topics in computer vision and pattern recognition. The objective of clustering is to discover the data structure and partition a group of data points into several clusters, where the similarity of data points within the same cluster is greater than the similarity from different clusters [1–8].

Structure of data is usually characterized by the affinity matrix of the graph whose edges denote the similarities between data points. If vertices belonging to each cluster are connected to be a component, i.e., there is no edge connecting between different clusters, cut is assigned to a value of zero in graph theory. Our purpose is to learn a graph with exact k number of components so that vertices in each connected component of the graph are partitioned into one cluster.

Given a cluster indicator matrix \mathbf{Z} , it can be constructed by data labels; $z_{ic} = 1$ if a data point \mathbf{x}_i is assigned to the c -th cluster, and $z_{ic} = 0$ otherwise. Since $\mathbf{Z}\mathbf{Z}^T$ is a strictly block diagonal matrix in the ideal case, many clustering methods

are designed to obtain a block diagonal affinity matrix or they use it as an important prior [9–13]. Actually, we find that a good quality of graph structure results in a good cluster indicator matrix, even if the affinity matrix is not a strictly block diagonal matrix. For example, in Figure 1, we can obtain an ideal cluster indicator matrix \mathbf{Z} with the graph itself because one connected component has exactly one cluster, but the affinity matrix \mathbf{W} (see Figure 1) constructed from the graph itself is not a strictly block diagonal matrix, i.e., some on-block diagonal elements are zeros. If the graph has k -connected components, we can directly obtain a good cluster indicator matrix with graph itself, even when the connected edges are relatively sparse in each connected component. It means that the graph structure rather than the strictly block diagonal affinity matrix is the intrinsic quality for obtaining a good clustering result.

We propose a novel graph-based clustering approach to exploit k -connected components for clustering, called k -component graph clustering (k -GC). Figure 1 shows schematic of the k -component graph learning. k -GC aims to learn a graph whose connected edges are tuned adaptively

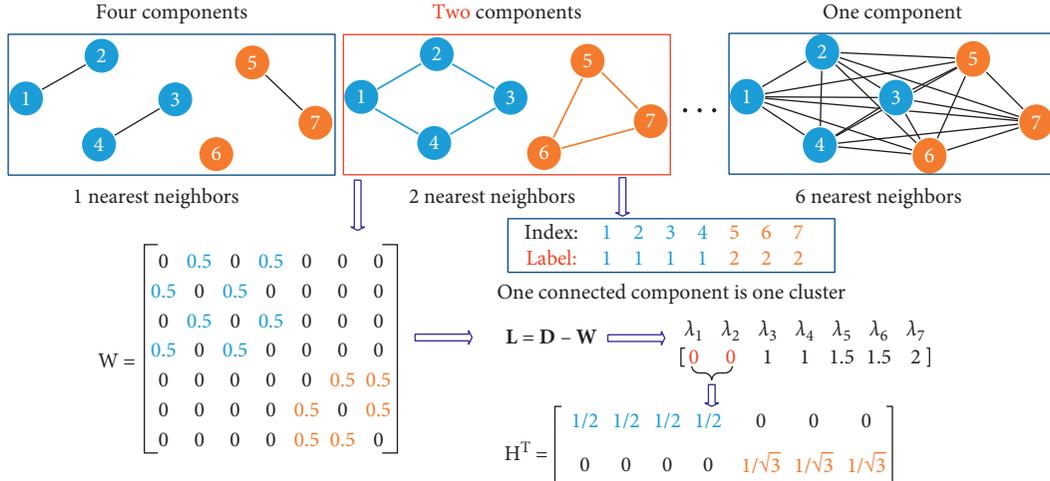


FIGURE 1: Schematic of the k -component graph learning.

until the graph has exactly k connected components [14]. To evaluate the effectiveness of k -GC, we have conducted experiments on six benchmark datasets in comparison to state-of-the-art approaches. Experimental results have well demonstrated that k -GC performs better than other approaches consistently. k -GC makes following contributions:

- (1) k -GC learns to obtain a graph with exact k -connected components. Since the vertices in each connected component belong to one cluster, labels are obtained directly from the learned graph itself.
- (2) The clustering indicators are obtained by using the learned graph itself without performing post-processing k -means clustering algorithm.
- (3) k -GC can be used as an alternative to spectral clustering (SC). Similar to SC, k -GC only needs to input an initial affinity graph without involving raw data.

2. Related Work

SC, which exploits the eigenstructure of a data affinity graph to partition data into different groups, has become one of the most fundamental clustering approaches. Standard SC uses the radial basis function to construct the affinity matrix [15], and its performance relies heavily on the eigenstructure of the affinity matrix [15–20].

In SC, the similarity between pairwise data points \mathbf{x}_i and \mathbf{x}_j is firstly computed by the radial basis function to construct the affinity matrix W , then the k number of eigenvectors of the normalized Laplacian matrix L corresponding to the top k smallest eigenvalues, $H = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_k] \in \mathbb{R}^{n \times k}$, are regarded as the low-dimensional embedding of raw data X , and k -means clustering is performed on H to obtain labels finally.

However, due to the ambiguity and uncertainty inherent in data structure, the intrinsic affinity matrix cannot be determined by a unified function. Since most existing SC methods use a predefined affinity matrix of the graph, the cut value is usually minimized but not zeroed, e.g., ratio cut [21],

normalized cut [16], and min-max cut [22], which results in a postprocessing k -means clustering algorithm to obtain the clustering labels [15, 17, 18].

Many approaches have been proposed to improve the performance of SC. Generally, they could be categorized into three paradigms:

- (1) How to improve data clustering using a predefined affinity matrix [12, 15, 23]
- (2) How to construct a better affinity matrix to obtain a better result than standard SC [17, 24, 25]
- (3) How to learn the affinity matrix and clustering structure simultaneously [26–29]

This paper is related to the third paradigm. In the third paradigm, objective functions of [26, 28, 29] usually involve the raw data but k -GC does not involve it. All of them employ a rank constraint on Laplacian matrix [26–29], and clustering with adaptive neighbors (CAN) [26] and constrained Laplacian rank (CLR) [27] are two methods more related to k -GC. CAN learns the graph from raw data. However, CLR learns the graph by minimizing the difference between the input initial graph and the learned graph.

In k -GC, the raw data are not involved in the objective function, and k -GC is directly modified from standard SC to tune the structure of graph so that an intrinsic graph with exact k -connected components is obtained.

3. k -Component Graph Clustering

Suppose that the data matrix is denoted by $X = [X_1, X_2, \dots, X_k] = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$, where X_c denotes the data matrix belonging to the c -th cluster, k denotes the cluster number, \mathbf{x}_i denotes the i -th data point, n is the number of data points, and d denotes the dimension.

3.1. Preprocessing. We propose a new simple neighborhood-preserving method to refine the raw data in this section. Assume that a data point \mathbf{x}_i and its nearest data point \mathbf{y}_i are both generated from a same point since they may be

disturbed by noise. Then, we define an equation to replace \mathbf{x}_i by the weighted linear combination of \mathbf{x}_i and \mathbf{y}_i :

$$\mathbf{x}_i \leftarrow \frac{\mathbf{x}_i + r_i \mathbf{y}_i}{1 + r_i}, \quad (1)$$

where r_i is defined by

$$r_i = \exp^{-\left(\|\mathbf{x}_i - \mathbf{y}_i\|_2^2\right)^{\sigma^2}}. \quad (2)$$

Since a data point is generally closer to the nearest one belonging to a same cluster than one to a different cluster, it is straightforward to check from equation (2) that r_i tends to 1 if \mathbf{x}_i and \mathbf{y}_i belong to a same cluster and r_i tends to 0 if \mathbf{x}_i and \mathbf{y}_i belong to different clusters. It implies that two data points in a same cluster become closer during iteration while two data points in different clusters have very small or even no influence on each other after using the preprocessing. With the iteration of equation (1), data points in same cluster become closer and closer so that the clustering is easier to carry out.

3.2. k -Component Graph. An undirected finite graph is denoted by G , and its vertex set is denoted by $V = \{v_1, v_2, \dots, v_n\}$. We consider the graph is weighted and assign a non-negative real weight w_{ij} to each pair vertices of v_i and v_j . The degree of a vertex $v_j \in V$ is defined by $d_j = \sum_{i=1}^n w_{ij}$. The Laplacian matrix of the graph G [30] is defined by

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \quad (3)$$

where \mathbf{D} is the diagonal matrix whose (j, j) -element is d_j .

For any vector $\mathbf{f} \in \mathbb{R}^n$, we have $\mathbf{f}^T \mathbf{L} \mathbf{f} = (1/2) \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \geq 0$, so \mathbf{L} is a positive semidefinite matrix. Since $\mathbf{L} \mathbf{1} = (\mathbf{D} - \mathbf{W}) \mathbf{1} = \mathbf{0}$, the smallest eigenvalue of \mathbf{L} is equal to zero, and the corresponding eigenvector is $\mathbf{1}$.

Theorem 1. *The number of connected components k is equal to the multiplicity of 0 as an eigenvalue of \mathbf{L} .*

Proof. Suppose there are k -connected components in graph G , the c -th subgraph is denoted by G_c , and the corresponding vertex set of G_c is denoted by V_c . Then, we have $\cup_{c=1}^k V_c = V$ and $V_i \cap V_{i \neq j} = \emptyset$. For an eigenvector \mathbf{h} of \mathbf{L} with eigenvalue zero, we have $0 = \mathbf{h}^T \mathbf{L} \mathbf{h} = (1/2) \sum_{i,j=1}^n w_{ij} (h_i - h_j)^2$. In the c -th connected component G_c , a connected edge is weighted by a positive value $w_{ij} > 0$, and the corresponding term of $(h_i - h_j)^2$ needs to be zeroed, i.e., h_i and h_j have to be a constant μ_c . $\forall v_i \in V_c$ and $\forall v_j \in V_c$, and the terms of $(h_i - h_j)^2$ need to be zeroed while these edges are connected with positive weights w_{ij} , which means that in the c -th component, $\forall v_i \in V_c$, and each h_i is of the same constant, i.e., $h_i = \mu_c$. These eigenvectors of different components are linearly independent, so the multiplicity of 0 as eigenvalue is the number of components of the graph G and vice versa.

Similar proof of Theorem 1 can be seen in previous works [13, 18, 31]. We explore Theorem 1 in the context of a specific example. In Figure 1, we show an intrinsic graph G with two connected components, and an intrinsic affinity

matrix \mathbf{W} of the graph G being consistent in Theorem 1 is given by

$$\mathbf{W} = \begin{bmatrix} 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \end{bmatrix}, \quad (4)$$

where we constrain the degree $d_j = \sum_{i=1}^n w_{ij} = 1$, i.e., the sum of each column of \mathbf{W} is equal to one.

The Laplacian matrix \mathbf{L} of equation (4) has an eigenvalue zero with the multiplicity $k = 2$, and we have the two corresponding eigenvectors \mathbf{h}_1 and \mathbf{h}_2 . They are concatenated by $H = [\mathbf{h}_1, \mathbf{h}_2]$, and its transposition is

$$\mathbf{H}^T = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix}. \quad (5)$$

If the graph G in Figure 1 corresponds to an affinity matrix \mathbf{W} in equation (4), i.e., the Laplacian matrix \mathbf{L} of equation (4) satisfies to Theorem 1, then the clustering labels are easy to obtain without further performing graph-cut or k -means clustering algorithms. Since vertices in each connected component of the graph are partitioned into one cluster, the clustering labels can be obtained easily by using the graph G with strongly connected-component algorithm [32].

Since it is a positive semidefinite matrix, \mathbf{L} has n non-negative eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. Theorem 1 indicates that if $\sum_{c=1}^k \lambda_c = 0$, then the graph G has k -connected components and the vertices are already partitioned into k clusters [18, 30, 31, 33]. Then, according to Fan's theorem [14, 34], we have

$$\sum_{c=1}^k \lambda_c = \min_{\mathbf{H}} \text{Tr}(\mathbf{H}^T \mathbf{L} \mathbf{H}) \quad (6)$$

$$\text{s.t. } \mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{H}^T \mathbf{H} = \mathbf{I},$$

where $\text{Tr}(\cdot)$ denotes the trace operator, $\mathbf{I} \in \mathbb{R}^{k \times k}$ is an identity matrix, $\mathbf{L} = \mathbf{D} - ((\mathbf{W}^T + \mathbf{W})/2)$ is the Laplacian matrix, and \mathbf{D} is a diagonal matrix whose diagonal elements are column sums of $((\mathbf{W}^T + \mathbf{W})/2)$.

The proof of Fan's theorem can be seen in [13, 35, 36].

It is straightforward to check that the objective function value of equation (6) does not generally tend to zero because the structure of graph G is varied by varying the graph construction methods. In the following section, we propose a method to tune the graph structure of G adaptively so that the objective function value of equation (6) tends to zero. \square

3.3. *k-GC Algorithm.* In this section, we will explore how to learn a graph \mathbf{W} with exact k -connected components [14].

The right term of equation (6) can be solved with respect to \mathbf{H} by the eigenvectors of \mathbf{L} , but it has trivial solution with respect to \mathbf{W} , i.e., only one w_{ij} is assigned to a value and others are zeroed in each column of \mathbf{W} . We add ℓ_2 -norm regularization to smooth the weights in \mathbf{W} . Then, we optimize \mathbf{W} and \mathbf{H} simultaneously:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{H}} \quad & \text{Tr}(\mathbf{H}^\top \mathbf{L} \mathbf{H}) + \beta \|\mathbf{W}\|_F^2 \\ \text{s.t.} \quad & \forall j, \mathbf{1}^\top \mathbf{w}_j = 1, \mathbf{w}_j \geq 0, \\ & \mathbf{H} \in \mathbb{R}^{n \times c}, \mathbf{H}^\top \mathbf{H} = \mathbf{I}, \end{aligned} \quad (7)$$

where β is the trade-off parameter.

If the affinity matrix is consistent in Theorem 1, the procedure of obtaining clustering labels is named as k -GC. In other words, we guarantee the objective function of standard SC to be zeroed in k -GC by means of tuning the graph structure. We divide the problem equation (7) into two subproblems and alternately solve them.

The first subproblem is to fix \mathbf{W} , updating \mathbf{H} . Then, equation (7) becomes

$$\begin{aligned} \min_{\mathbf{H}} \quad & \text{Tr}(\mathbf{H}^\top \mathbf{L} \mathbf{H}) \\ \text{s.t.} \quad & \mathbf{H} \in \mathbb{R}^{n \times c}, \mathbf{H}^\top \mathbf{H} = \mathbf{I}. \end{aligned} \quad (8)$$

Equation (8) is solved by calculating the eigenvectors of \mathbf{L} .

The second subproblem is to fix \mathbf{H} , updating \mathbf{W} . Then, equation (7) becomes

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{i,j=1}^n \|\mathbf{h}^{(i)} - \mathbf{h}^{(j)}\|_2^2 w_{ij} + \beta \|\mathbf{W}\|_F^2 \\ \text{s.t.} \quad & \forall j, \mathbf{w}_j \geq 0, \mathbf{1}^\top \mathbf{w}_j = 1, \end{aligned} \quad (9)$$

where $\mathbf{h}^{(i)}$ denotes the i -th column of \mathbf{H}^\top .

Each column of \mathbf{W} is independent, so solving equation (9) is equal to optimizing the following problem:

$$\begin{aligned} \min_{\mathbf{w}_j} \quad & \sum_{i=1}^n \|\mathbf{h}^{(i)} - \mathbf{h}^{(j)}\|_2^2 w_{ij} + \beta \sum_{i=1}^n w_{ij}^2 \\ \text{s.t.} \quad & \mathbf{w}_j \geq 0, \mathbf{1}^\top \mathbf{w}_j = 1. \end{aligned} \quad (10)$$

$\|\mathbf{h}^{(i)} - \mathbf{h}^{(j)}\|_2^2$ is denoted by g_i and $\mathbf{h}^{(j)}$ is a fixed vector when we solve the j -th column \mathbf{w}_j . Solving equation (10) is equal to optimizing the following problem:

$$\begin{aligned} \min_{\mathbf{w}_j} \quad & \frac{1}{2} \left\| \mathbf{w}_j + \frac{1}{2\beta} \mathbf{g} \right\|_2^2 \\ \text{s.t.} \quad & \mathbf{w}_j \geq 0, \mathbf{1}^\top \mathbf{w}_j = 1. \end{aligned} \quad (11)$$

where $\mathbf{g} = [g_1, g_2, \dots, g_n]^\top$ is a constant vector.

Equation (11) is a Euclidean projection problem on the simplex space, and there are several algorithms for solving it [37–39]. According to the Karush–Kuhn–Tucker condition [37], it can be verified that the optimal solution \mathbf{w}_j^* is

$$\mathbf{w}_j^* = \left(-\frac{\mathbf{g}}{2\beta} + \eta \right)_+. \quad (12)$$

Equation (10) will be analysed specifically in Section 3.4, and we show how to obtain exact k -connected components.

We alternately optimize equations (9) and (8) until the sum of the top k smallest eigenvalues of \mathbf{L} becomes zero. The algorithm for solving equation (7) is summarized by Algorithm 1.

3.4. *k-Component Graph Learning.* We find a trade-off between two graph structures: the first case is that one vertex is connected with only one other vertex in vertex set V and the second case is that all vertices are connected with each other by the same weight $1/n$. The trade-off renders the objective function of equation (6) close to zero.

We optimize each column of \mathbf{W} independently, and one column of \mathbf{W} is denoted by \mathbf{w}_j . Values in the vector \mathbf{w}_j are zero or positive: $w_{ij} = 0$ means \mathbf{v}_i has no edge with \mathbf{v}_j and $w_{ij} > 0$ means \mathbf{v}_i and \mathbf{v}_j are connected with an edge.

The first is to optimize the following:

$$\begin{aligned} \min_{\mathbf{w}_j} \quad & \mathbf{w}_j^\top \mathbf{g} \\ \text{s.t.} \quad & \mathbf{w}_j \geq 0, \mathbf{1}^\top \mathbf{w}_j = 1. \end{aligned} \quad (13)$$

It returns the minimum value $g_i = \min(\mathbf{g})$, i.e., \mathbf{v}_j is only connected to the other \mathbf{v}_i .

The second is the following:

$$\begin{aligned} \min_{\mathbf{w}_j} \quad & \mathbf{w}_j^\top \mathbf{w}_j \\ \text{s.t.} \quad & \mathbf{w}_j \geq 0, \mathbf{1}^\top \mathbf{w}_j = 1. \end{aligned} \quad (14)$$

It returns $w_{ij}^* = 1/n, \forall j$.

Thus, an adaptive graph learning objective function is

$$\begin{aligned} \min_{\mathbf{w}_j} \quad & \mathbf{w}_j^\top \mathbf{g} + \beta_j \mathbf{w}_j^\top \mathbf{w}_j \\ \text{s.t.} \quad & \mathbf{w}_j \geq 0, \mathbf{1}^\top \mathbf{w}_j = 1, \end{aligned} \quad (15)$$

where β_j is the trade-off parameter.

If $\beta_j = 0$, then optimizing equation (15) is equivalent to solving equation (13); if β_j tends to an infinity positive value, then equation (15) becomes equation (14). If we want to learn a sparse graph, we can tune β_j to be a small value; if we want to learn a graph with more edges, we can tune β_j to be a relatively large value.

Solving equation (15) is equal to optimizing the problem:

$$\begin{aligned} \min_{\mathbf{w}_j} \quad & \frac{1}{2} \left\| \mathbf{w}_j + \frac{1}{2\beta_j} \mathbf{g} \right\|_2^2 \\ \text{s.t.} \quad & \mathbf{w}_j \geq 0, \mathbf{1}^\top \mathbf{w}_j = 1. \end{aligned} \quad (16)$$

According [40], the optimal affinities w_{ij}^* are given by

$$w_{ij}^* = \begin{cases} \frac{g_{m+1} - g_i}{mg_{m+1} - \sum_{i=1}^m g_i}, & \text{if } i \leq m; \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

- (1) **Input:** A data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, the number of neighbors m , and the number of clusters k .
- (2) **Output:** The affinity graph \mathbf{W} .
- (3) **Initialize:** The graph \mathbf{W} is initialized by equation (9) where we replace \mathbf{H} by \mathbf{X}^\top ; \mathbf{H} is initialized with initial graph \mathbf{W} by solving equation (8); and $\beta = \sum_{j=1}^n \beta_j$.
- (4) Iterate equation (1) several times.
- (5) **While** not converge **do**
- (6) **For** $j \in \{1, 2, \dots, n\}$ **do**
- (7) Update \mathbf{w}_j by using equation (12);
- (8) **end for**
- (9) $\mathbf{W} = (\mathbf{W} + \mathbf{W}^\top)/2$;
- (10) Update \mathbf{H} by using equation (8), i.e., \mathbf{H} is formed by k eigenvectors with the top k smallest eigenvalues of \mathbf{L} .
- (11) **end while**

ALGORITHM 1: k -GC algorithm.

Actually, it is straightforward to check from equation (17) that the number of neighbors m is also determined by the setting of β_j . In practice, the structure of the graph G can be tuned coarsely by m and can be tuned finely by the trade-off parameter β_j . m has an explicit meaning while β_j has an implicit relation with the structure of the graph. We tune both of them to obtain an intrinsic affinity matrix.

As standard SC, we can input any affinity matrix for k -GC. There is one parameter β in equation (7). In Algorithm 1, we use Eq. (9) to obtain the initial graph by replacing \mathbf{H} with \mathbf{X}^\top . For each \mathbf{w}_j , $j \in \{1, 2, \dots, n\}$, we have different β_j , so before the iteration, we set $\beta = \sum_{j=1}^n \beta_j$ for preserving the initial structure. The initial graph structure is mainly determined by m .

3.5. Convergence Analysis. Since the second-order derivative of equation (16) with respect \mathbf{w}_j is equal to $1 \geq 0$, equation (16) is a convex problem. Because the Laplacian matrix is a positive semidefinite matrix, equation (8) is a convex optimization problem. Optimizing \mathbf{W} and \mathbf{H} alternately, both of them decrease monotonically. As a result, the overall objective function value of equation (7) decreases monotonically in each iteration until Algorithm 1 converges.

3.6. Computational Complexity Analysis. The first step of the objective function equation (7) is to solve equation (10). We need $O(n)$ time to compute $\mathbf{h}^{(j)}$, and $O(t_1 n)$ to solve equation (15) where t_1 is the iteration number. n times are needed to calculate each \mathbf{w}_j , $\forall j$, so the complexity of the first step of equation (7) is $O((t_1 + 1)n^2)$. The second step is an eigendecomposition procedure, and the complexity of the generalized eigenvector problem is $O((n + k)n^2)$. For solving equation (8), we need to calculate the k eigenvectors of \mathbf{L} , so its cost is $O(kn^2)$. Thus, the total complexity of equation (7) is

$$O(((t_1 + 1 + k)n^2)t_o), \quad (18)$$

where t_o is the number of iterations of the two steps.

4. Experimental Results

In this section, we conduct experiments on six datasets to demonstrate the effectiveness of k -GC in terms of clustering accuracy (ACC) and normalized mutual information (NMI).

4.1. Dataset Description. Six datasets include the following:

- (1) Two-moon dataset is a randomly generated synthesis dataset and has two clusters of data distributed in the moon shape. Each cluster has 100 samples and the noise percentage is set to 0.20.
- (2) Path-based dataset [41] is made up of 300 samples which belong to three clusters.
- (3) Yale dataset (<http://vision.ucsd.edu/content/yale-face-database>) contains 165 grayscale images of 15 individuals. Each individual has 11 images taken under different facial expression or configuration.
- (4) ORL dataset (<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>) contains ten different images of each with 40 distinct subjects. For some subjects, the images were taken at different time, varying the lighting, facial expressions, and facial details. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position.
- (5) COIL-20 dataset [42] contains 1440 images of 20 object categories. Each category includes 72 images and all the images are normalized to 32×32 pixel arrays.
- (6) Notting-Hill dataset is derived from the movie "Notting Hill" and includes 4660 faces in 76 tracks belonging to five main casts.

The first two datasets are two-dimensional synthesis datasets. For datasets of Yale, ORL, COIL-20, and Notting-Hill, we employ the intensity feature of these images [43, 44]. A small fraction of samples in these image datasets is shown in Figure 2. Dataset description is summarized in Table 1.

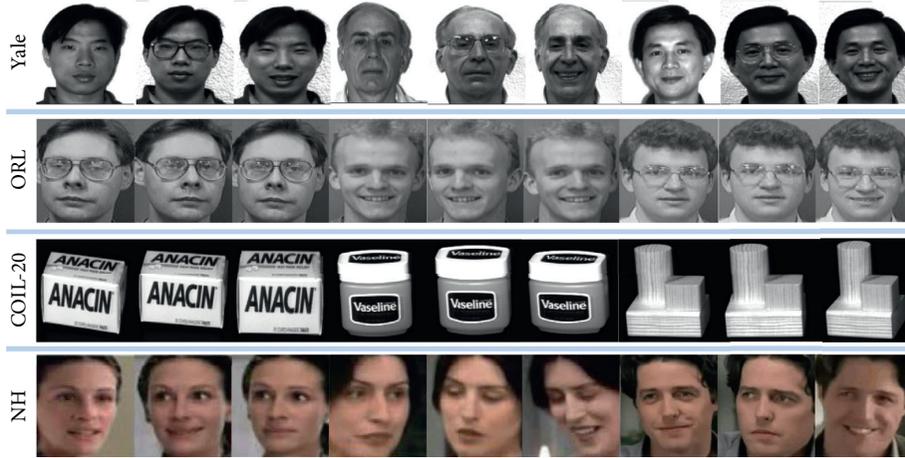


FIGURE 2: Example images of different datasets.

TABLE 1: Dataset description.

Dataset	Samples	Dimensions	Clusters
Two-moon	200	2	2
Path-based	300	2	3
Yale	165	4096	15
ORL	400	4096	40
COIL-20	1440	1024	20
Notting-Hill	4660	2000	5

4.2. *Experimental Setup.* We compare k -GC with six methods, and the results are summarized in Table 2. We select three classic methods: k -means clustering [45], ratio cut clustering (R-cut) [21], and self-tuning SC (ST-SC) [17]; besides, we select three related state-of-the-art methods: robust graph for SC (RG-SC) [24], clustering with adaptive neighbors (CAN) [26], and constrained Laplacian rank (CLR) [27].

The default parameters given by the respective authors are adopted.

- (1) k -means [45] clusters samples based on the similarity between samples and cluster centroids. There is no parameter to tune in k -means. We run k -means clustering 10 times to evaluate the performance. In order to reduce the effect of random initialization, we run 30 times and report the result with the minimum value of the objective function of k -means among these results in each time.
- (2) R-cut [21] finds the first k eigenvectors of the Laplacian matrix of the graph so as to minimize the similarity between two parts in the graph. Specifically, R-cut advocates the second eigenvalue of the Laplacian matrix which gives the optimal solution.
- (3) ST-SC [17] constructs a graph in a local scale and each data point chooses different neighbors. It has a parameter to determine the number of nearest neighbors for graph construction, and we tune the neighbor number to report the best results in terms of the objective value of k -means.

- (4) RG-SC [24] generates robust affinity graphs via identifying and exploiting discriminative features for spectral clustering based on the unsupervised clustering random forest. The default parameters are used in our experiments.
- (5) CAN [26] learns the data similarity matrix and clustering structure simultaneously and uses a rank constraint to create the clustering structure in the similarity matrix as several disconnected components. The number of nearest neighbors is searched from 2 to 50 with incremental 2.
- (6) CLR [27] imposes a Laplacian rank constraint on the learned graph which best approximates the input initial affinity graph. In our experiments, the number of nearest neighbors is searched from 2 to 50 with interval 2 to obtain the best parameter.

For k -GC, we tune the parameter m in the range of [5, 40] with interval two, and β is fixed by $\beta = \sum_{j=1}^n \beta_j$. For pre-processing, we iterate equation (1) ten times with $\sigma = 12$. We select optimal m in terms of objective value of k -means clustering performed on \mathbf{H} . k -GC obtains the clustering results from the learned graph itself because data points in each connected component belong to one cluster, and the clustering indicators are directly obtained according to Tarjan's strongly connected-component algorithm [32]. In practice, we can determine β in a heuristic way to accelerate the procedure [26]. As the iteration stopping condition of the sum of the top k smallest eigenvalues of \mathbf{L} being zeroed, i.e., the first term of equation (7) is zeroed, is stronger than conventional iteration stopping condition [26], we adopt the former. Because of the iteration stopping condition, we set β to a constant (e.g., $\beta = \sum_{j=1}^n \beta_j$), then increase β if the connected components of \mathbf{W} are larger than k , and decrease β if they are smaller than k during iteration.

For all these methods, we run each method 10 times and report the mean of performance as well as the standard deviation in Table 2.

4.3. *Results and Analysis.* We choose six datasets to demonstrate the effectiveness of k -GC, the first two of which are synthetic while the rest are real-world datasets. In addition,

TABLE 2: Clustering performance.

Methods	<i>k</i> -means	R-cut	ST-SC	RG-SC	CAN	CLR	<i>k</i> -GC
<i>ACC</i>							
Two-moon	75.50 ± 0.00	73.00 ± 0.00	79.00 ± 0.00	78.00 ± 0.00	93.00 ± 0.00	87.50 ± 0.00	98.50 ± 0.00
Path-based	74.33 ± 0.00	76.85 ± 2.05	79.33 ± 0.00	60.66 ± 0.00	87.00 ± 0.00	87.00 ± 0.00	99.00 ± 0.00
Yale	50.30 ± 2.68	29.75 ± 1.86	57.63 ± 1.49	62.90 ± 0.25	52.73 ± 0.00	59.39 ± 0.00	68.48 ± 0.00
ORL	55.45 ± 3.10	30.12 ± 1.37	68.27 ± 1.78	59.82 ± 0.86	56.50 ± 0.00	63.75 ± 0.00	70.75 ± 0.00
COIL-20	66.79 ± 2.65	45.72 ± 0.99	69.34 ± 0.06	76.65 ± 0.02	90.14 ± 0.00	87.29 ± 0.00	94.03 ± 0.00
Notting-Hill	79.53 ± 0.61	48.92 ± 0.00	78.71 ± 0.00	80.86 ± 0.01	70.62 ± 0.00	74.79 ± 0.00	83.58 ± 0.00
<i>NMI</i>							
Two-moon	19.79 ± 0.00	15.92 ± 0.00	25.85 ± 0.00	24.10 ± 0.00	69.37 ± 0.00	54.88 ± 0.00	90.22 ± 0.00
Path-based	51.28 ± 0.01	50.92 ± 8.38	57.29 ± 0.00	27.72 ± 2.35	75.63 ± 0.00	75.63 ± 0.00	95.82 ± 0.00
Yale	56.17 ± 1.98	36.27 ± 2.62	61.72 ± 0.92	65.49 ± 0.27	55.40 ± 0.00	59.21 ± 0.00	68.63 ± 0.00
ORL	74.31 ± 1.56	55.26 ± 1.04	82.60 ± 0.72	77.91 ± 0.43	74.07 ± 0.00	79.98 ± 0.00	83.33 ± 0.00
COIL-20	77.83 ± 1.22	55.99 ± 0.47	78.48 ± 0.30	84.83 ± 0.03	94.60 ± 0.00	94.14 ± 0.00	96.42 ± 0.00
Notting-Hill	68.09 ± 1.08	26.94 ± 0.00	64.19 ± 0.00	67.39 ± 0.01	61.80 ± 0.00	71.05 ± 0.00	77.93 ± 0.00

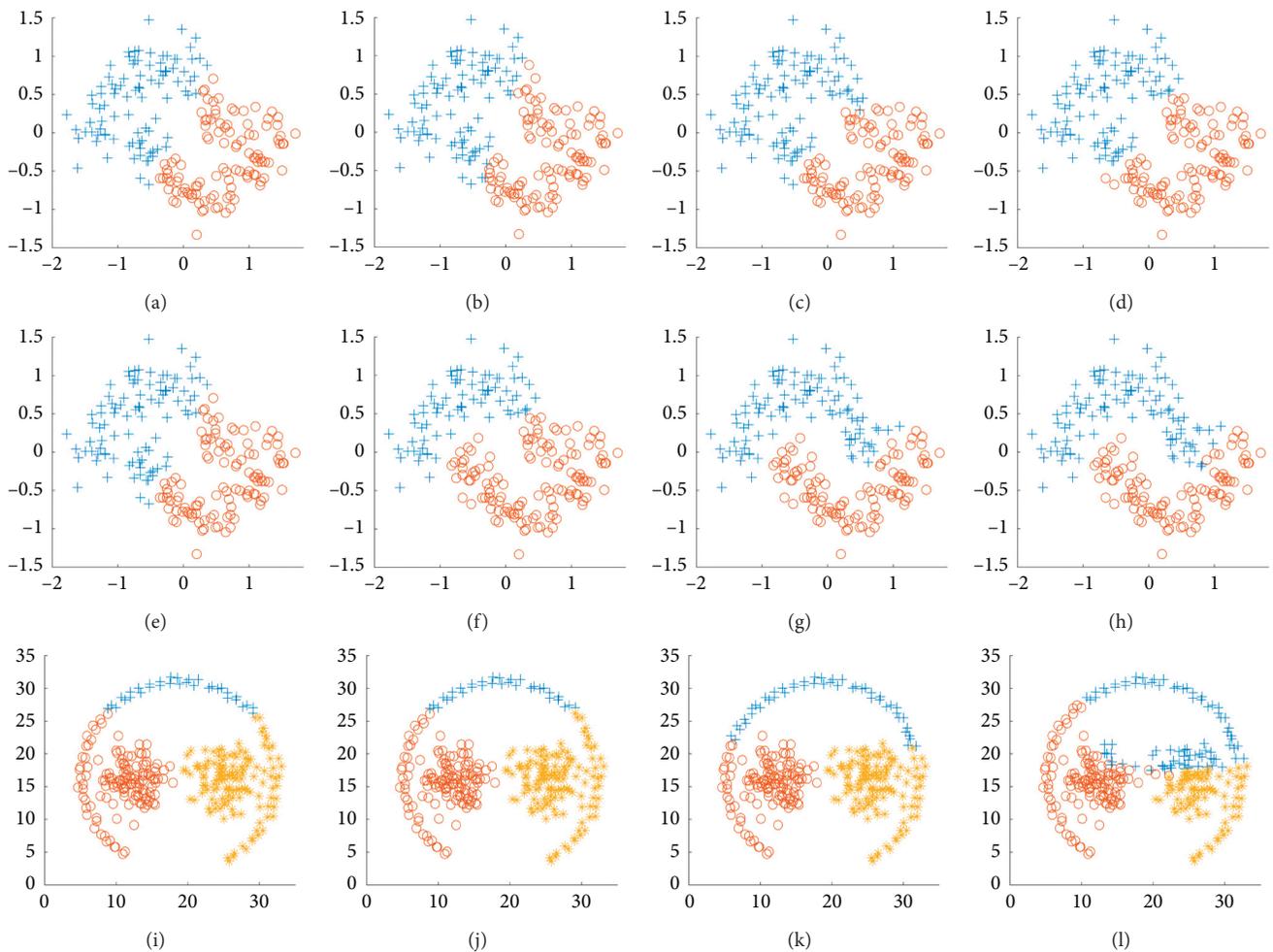


FIGURE 3: Continued.

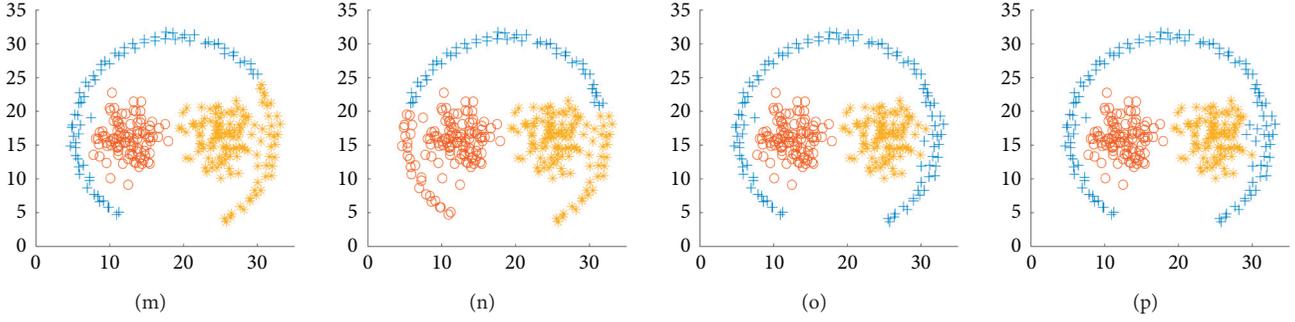


FIGURE 3: Results of comparing methods on two-moon and path-based datasets. GT denotes the ground truth. (a) k -means, (b) R-cut, (c) ST-SC, (d) RG-SC, (e) CAN, (f) CLR, (g) k -GC, (h) GT, (i) k -means, (j) R-cut, (k) ST-SC, (l) RG-SC, (m) CAN, (n) CLR, (o) k -GC, and (p) GT.

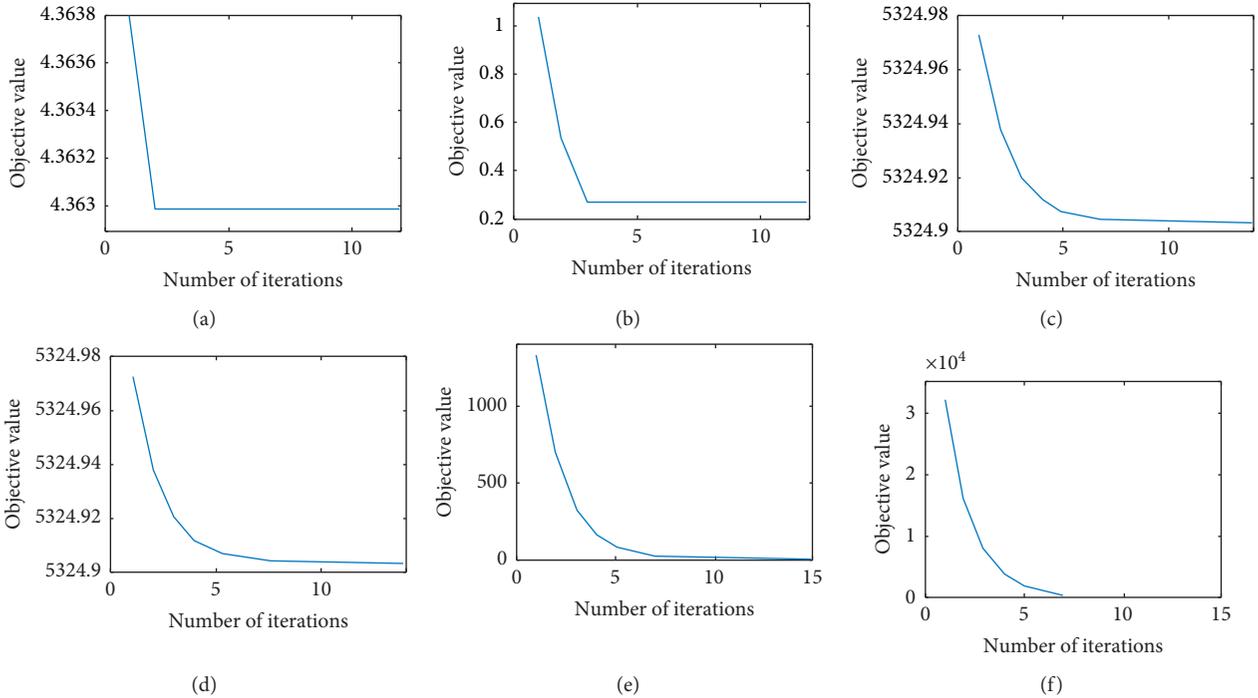


FIGURE 4: Convergence behaviors of k -GC. (a) Two-moon dataset. (b) Path-based dataset. (c) Yale dataset. (d) ORL dataset. (e) COIL-20 dataset. (f) Notting-Hill dataset.

quantitative results of different methods are shown in Table 2, from which we can find that we have improved the performance to a large extent in comparison to the state-of-the-art methods. The clustering results of the comparing methods using the two synthetic datasets are visualized in Figure 3 so that we can observe in an intuitive way that k -GC approximates the ground truth mostly. What is notable is that on two synthetic datasets of two-moon and path-based, we have a surprisingly good result approximating to 100%, meaning almost entirely accurate clustering. This implies the great importance of the intrinsic graph structure with a comprehensive analysis of which we obtain a significantly success over other methods.

Figure 4 shows that the objective value of k -GC, equation (7), is nonincreasing during iterations. They all

converge to a fixed value very fast in 10 iteration times. k -GC only needs several times to iterate alternately to obtain a cut-zeroed graph while the conventional SC-based methods need to perform postprocessing by using k -means clustering algorithm. Therefore, the k -GC algorithm is effective overall.

5. Conclusion

In this paper, we have proposed a novel k -component graph clustering method, which learns to obtain a graph with exact k -connected components. Since the vertices in each connected component of the intrinsic graph belong to one cluster, labels are obtained directly from the learned graph itself without performing further graph-cut or k -means clustering

algorithms. k -GC learns the affinity matrix and clustering structure simultaneously. Moreover, k -GC can serve as an alternative to SC due to its simplicity and the effectiveness over the standard SC. This paper focuses on the spectral analysis of the graph Laplacian, to which the sum of top k smallest eigenvalues is zeroed. It renders the graph structure update until it has exactly k number of components. The efficient algorithm as well as the optimization algorithm is presented after plentiful analysis. Experiments on six benchmarks have demonstrated the superiority of k -GC.

Data Availability

All the datasets are available at https://github.com/lzu-cvpr/data_for_clustering.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the 13th Five-Year Informatization Plan of the Chinese Academy of Sciences (grant nos. XXH13506 and XXH13505-220) and Data Sharing Fundamental Program for Construction of the National Science and Technology Infrastructure Platform (grant no. Y719H71006).

References

- [1] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on elite selection of weighted clusters," *Advances in Data Analysis and Classification*, vol. 7, no. 2, pp. 181–208, 2013.
- [2] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm," *Pattern Analysis and Applications*, vol. 18, no. 1, pp. 87–112, 2015.
- [3] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1617–1632, 2016.
- [4] X. Chang, Z. Ma, M. Lin, Y. Yang, and A. G. Hauptmann, "Feature interaction augmented sparse learning for fast kinect motion detection," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3911–3920, 2017.
- [5] X. Chang, Z. Ma, Y. Yang, Z. Zeng, and A. G. Hauptmann, "Bi-level semantic representation analysis for multimedia event detection," *IEEE Transactions on Cybernetics*, vol. 47, no. 5, pp. 1180–1197, 2016.
- [6] X. Chang and Y. Yang, "Semisupervised feature analysis by mining correlations among multiple tasks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2294–2305, 2016.
- [7] Z. Li, F. Nie, X. Chang, and Y. Yang, "Beyond trace ratio: weighted harmonic mean of trace ratios for multiclass discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2100–2110, 2017.
- [8] A. Almagbile, "Estimation of crowd density from uavs images based on corner detection procedures and clustering analysis," *Geo-spatial Information Science*, vol. 22, no. 1, pp. 23–34, 2019.
- [9] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.
- [10] E. Elhamifar and R. Vidal, "Sparse subspace clustering: algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.
- [11] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *Proceedings of the CVPR*, pp. 3818–3825, IEEE, Columbus, OH, USA, June 2014.
- [12] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: single-view to multi-view," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2833–2843, 2016.
- [13] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1261–1270, 2019.
- [14] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Transactions on Cybernetics*, vol. 48, no. 10, pp. 2887–2895, 2018.
- [15] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Proceedings of the NIPS*, vol. 14, pp. 849–856, Vancouver, Canada, December 2002.
- [16] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [17] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proceedings of the NIPS*, pp. 1601–1608, Vancouver, Canada, December 2005.
- [18] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [19] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 1984–1993, 2018.
- [20] K. Zhan, X. Chang, J. Guan, L. Chen, Z. Ma, and Y. Yang, "Adaptive structure discovery for multimedia analysis using multiple features," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1826–1834, 2019.
- [21] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074–1085, 1992.
- [22] C. H. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Proceedings of the ICDM*, pp. 107–114, IEEE, San Jose, CA, USA, November–December 2001.
- [23] T. Xiang and S. Gong, "Spectral clustering with eigenvector selection," *Pattern Recognition*, vol. 41, no. 3, pp. 1012–1029, 2008.
- [24] X. Zhu, C. Change Loy, and S. Gong, "Constructing robust affinity graphs for spectral clustering," in *Proceedings of the CVPR*, pp. 1450–1457, IEEE, Columbus, OH, USA, June 2014.
- [25] J. Huang, F. Nie, and H. Huang, "A new simplex sparse learning model to measure data similarity for clustering," in *Proceedings of the IJCAI*, pp. 3569–3575, Buenos Aires, Argentina, July 2015.
- [26] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proceedings of the KDD*, vol. 20, pp. 977–986, ACM, New York, NY, USA, August 2014.

- [27] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Proceedings of the AAAI*, AAAI, Phoenix, AZ, USA, pp. 1969–1976, February 2016.
- [28] X. Zhu, W. He, Y. Li et al., "One-step spectral clustering via dynamically learning affinity matrix and subspace," in *Proceedings of the AAAI*, pp. 2963–2969, San Francisco, CA, USA, February 2017.
- [29] Z. Kang, C. Peng, and Q. Cheng, "Twin learning for similarity and clustering: a unified kernel approach," in *Proceedings of the AAAI*, pp. 2080–2086, San Francisco, CA, USA, February 2017.
- [30] F. R. Chung, *Spectral Graph Theory*, American Mathematical Society, Providence, Rhode Island, 1997.
- [31] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The Laplacian spectrum of graphs," *Graph Theory, Combinatorics, and Applications*, vol. 2, no. 12, pp. 871–898, 1991.
- [32] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.
- [33] B. Mohar, "Some applications of laplace eigenvalues of graphs," in *Graph Symmetry*, G. Hahn and G. Sabidussi, Eds., pp. 225–275, Kluwer Academic Publishers, Berlin, Germany, 1997.
- [34] K. Fan, "On a theorem of Weyl concerning eigenvalues of linear transformations I," *Proceedings of the National Academy of Sciences*, vol. 35, no. 11, pp. 652–655, 1949.
- [35] M. L. Overton and R. S. Womersley, "On the sum of the largest eigenvalues of a symmetric matrix," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 1, pp. 41–45, 1992.
- [36] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, "Spectral K-way ratio-cut partitioning and clustering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 9, pp. 1088–1096, 1994.
- [37] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [38] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1 -ball for learning in high dimensions," in *Proceedings of the ICML*, vol. 25, ACM, Helsinki, Finland, pp. 272–279, July 2008.
- [39] L. Condat, "Fast projection onto the simplex and the ℓ_1 ball," *Mathematical Programming*, vol. 158, no. 1-2, pp. 575–585, 2016.
- [40] C. Chen, C. Niu, X. Zhan, and K. Zhan, "Generative approach to unsupervised deep local learning," *Journal of Electronic Imaging*, vol. 28, no. 4, Article ID 043005, 2019.
- [41] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191–203, 2008.
- [42] S. A. Nene, S. K. Nayar, and H. Murase, *Columbia object image library (COIL-20)*, Columbia University, New York, NY, USA, 1996.
- [43] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proceedings of the ICCV*, pp. 1582–1590, Santiago, Chile, December 2015.
- [44] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proceedings of the CVPR*, pp. 586–594, Santiago, Chile, December 2015.
- [45] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, pp. 281–297, Oakland, CA, USA, January 1967.