Bipartite Graph based Multi-view Clustering

Lusi Li and Haibo He, *Fellow, IEEE*

Abstract—For graph-based multi-view clustering, a critical issue is to capture consensus cluster structures via a two-stage learning scheme. Specifically, first learn similarity graph matrices of multiple views and then fuse them into a unified superior graph matrix. Most current methods learn pairwise similarities between data points for each view independently, which is widely used in single view. However, the consensus information contained in multiple views are ignored, and the involved biases lead to an undesirable unified graph matrix. To this end, we propose a bipartite graph based multi-view clustering (BIGMC) approach. The consensus information can be represented by a small number of representative uniform anchor points for different views. A bipartite graph is constructed between data points and the anchor points. BIGMC constructs the bipartite graph matrices of all views and fuses them to produce a unified bipartite graph matrix. The unified bipartite graph matrix in turn improves the bipartite graph similarity matrix of each view and updates the anchor points. The final unified graph matrix forms the final clusters directly. In BIGMC, an adaptive weight is added for each view to avoid outlier views. A low-rank constraint is imposed on the Laplacian matrix of the unified matrix to construct a multi-component unified bipartite graph, where the component number corresponds to the required cluster number. The objective function is optimized in an alternating optimization fashion. Experimental results on synthetic and real-world data sets demonstrate its effectiveness and superiority compared with the state-of-the-art baselines.

Index Terms—Multi-view clustering, similarity matrix, consensus information, bipartite graph.

1 Introduction

Clustering has long been serving as a critical unsupervised technique in pattern recognition, data mining, and machine learning. The aim of clustering is to group data objects into clusters such that data objects in the same cluster are more similar than those in different clusters. However, most existing clustering methods are concerned about single-view learning [1]. As Internet and communication technologies develop rapidly, many real-world data can be extracted from multiple sources [2, 3], which makes it possible to produce multi-view data. In multi-view data, each object is associated with much richer information [4]. How to make full use of the information contained in multiple views to improve clustering results is referred to as be multi-view clustering [5].

Obviously, each view has its biases. If the multi-view clustering algorithms cannot explore valuable information and cope appropriately with multiple views, the clustering performance may be poorer than that by single-view clustering methods [6]. Thus, compared with single-view clustering, multi-view clustering is expected to achieve more robust and precise clustering results via exploiting the complementary information in multiple views [7]. Three main challenges need to overcome. The first one is how to extract the valuable information from multiple views [8]. The second one is how to integrate these extracted information effectively [9, 10]. The third one is how to learn the importance of each view for the clustering task [11]. Note that these three issues should be figured out simultaneously. To this end, a variety of multi-view clustering methods have been proposed. Among existing studies, graph-based methods are representative [12, 13]. The structures of graphs consist of sets of vertexes and weighted edges among them. The similarity between any two vertexes is represented by the weight associated with the edge that connects them. Hence, graphs can effectively express the relationships among various types of data objects [14]. In graphs, each vertex corresponds to one data object and each weighted edge represents the similarity relationship between two objects it connects.

In practice, the similarity relationships are expressed differently in different views [15]. Graph-based multi-view clustering methods aim to encode the similarity relationships among the data objects in the form of a unified graph matrix by combining the graph matrices of all views [16]. For the unified graph matrix, each non-zero element indicates the complementary similarity between two data objects. The final clusters are formed by employing an additional clustering method on the unified graph matrix. The clustering performance depends on the quality of each view graph and the fusion strategy. Although they have achieved some successes, there still exist several limitations. First, the consensus information of different views are not considered when learning each view graph matrix. Most existing methods learn pairwise similarities between objects for each view independently. This often leads to that the involved biases affect the quality of each view graph matrix. Motivated by [17], our method captures the consensus information by learning a small number of representative uniform anchor points for different views. Each anchor point is the centroid of the corresponding sub-cluster. That is to say, each view has an anchor set and these anchor points in different views preserve the information within the same sub-clusters. Second, they keep both the pre-given anchor set and the learned view graph matrix fixed in the fusion process (e.g., [17]). In this case, they are sensitive to the initialization and easy to trap in local optimum. Our method

This work was supported by the National Science Foundation under grant ECCS 1917275. (Corresponding author: H. He).

L. Li and H. He are with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: {lusi_li, haibohe}@uri.edu).

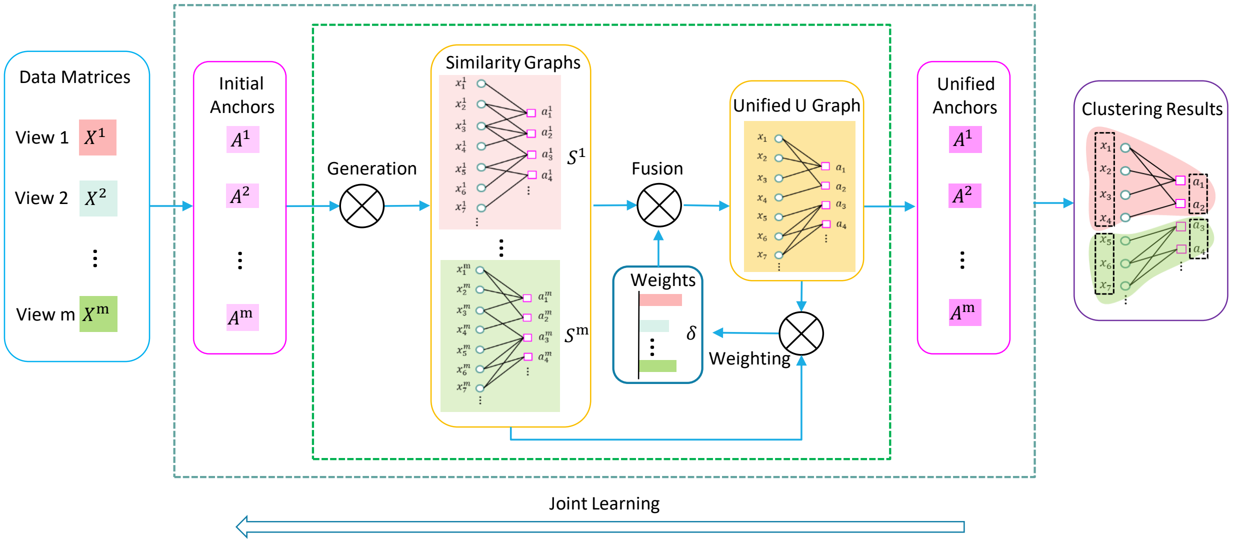


Fig. 1. The framework of our proposed BIGMC.

learns each view bipartite graph matrix, the unified graph matrix, and uniform anchor points jointly in a mutually reinforcing way. Each of them can help the learning of the others. Third, most of them cannot adaptively learn the weight of each view without an additive hyper-parameter. The optimal value of the additive hyper-parameter needs to search in a large range [18]. Our method can determine an optimal weight for each view adaptively based on the corresponding learned view bipartite graph and the unified graph matrix.

To address these limitations simultaneously, we propose a novel multi-view clustering approach, denoted by *BIpartite Graph-based Multi-view Clustering* (BIGMC). The overall framework of BIGMC is shown in Fig. 1. To be specific, from the input of multi-view data matrices, we create t initial uniform anchor points for different views denoted as A. Then the graph of each view is generated based on the similarity between data points and the anchor points, which is referred to as “data-to-anchor" similarity graph and denoted as S. Afterwards, all Ss from multiple views are employed to learn a unified graph matrix U in the fusion procedure. At the meantime, a weight for each view (δ) is added adaptively based on Ss and U indicating its importance. A low rank constraint is imposed on the Laplacian matrix of a unified bipartite graph associated with U, which aims to constrain that the bipartite graph has c number of connected components corresponding to the required number of clusters. Next, the obtained unified matrix U would go back to improve the Ss and δ of each view until convergence. According to the converged unified graph matrix U, we can get the unified anchor points A for each view. If they are different from the initial anchor points, we would improve all As to in turn update the Ss, the unified graph matrix U, and the weight δ until they are identical. The final clusters are formed directly based on U.

Hence, the main contributions can be summarized as follows.

- We propose a novel bipartite graph based multi-view clustering (BIGMC) approach. BIGMC can learn and make good use of the consensus information represented by a small number of uniform anchor points, which alleviates the influence of biases contained in multiple views.
- BIGMC jointly learns the similarity bipartite graph for each view, the unified bipartite graph, and the consensus anchors in a mutually reinforcing way. It can also determine the weight for each bipartite graph automatically without introducing an additive hyper-parameter. The final clusters are generated directly based on the unified bipartite graph when the anchors are identical in different views.
- BIGMC employs an efficient alternating iterative optimization strategy to solve the variable optimization problem step by step, where each sub-problem has an optimal solution.
- Experimental results on both synthetic and real-world data sets demonstrate the effectiveness of the proposed BIGMC and the superiority than the state-of-the-art baselines.

The rest of this paper is organized as follows. Section 2 gives a brief introduction of related multi-view clustering methods. Section 3 presents the proposed bipartite graph based multi-view clustering approach. The optimization strategy of this problem is given in Section 4. Extensive experiments are shown in Section 5. At last, Section 6 concludes this paper.

2 Related Work

Numerous multi-view clustering approaches have been proposed, and can be roughly divided into four categories based on different learning strategies: co-training learning [19, 20], multi-kernel learning [21, 22], subspace learning [23, 24], and spectral learning [25, 26]. Among them, the co-training learning aims to produce a learner for each view by using the learned knowledge from one another for the partitions of different views. The authors in [20] combine linear discriminant analysis and K-means method [27] with co-training into a unified framework. The main idea of multi-kernel learning is to combine multiple kernels in a linearly or non-linearly manner to perform multi-view clustering, where a base kernel is predefined for each view. The work in [21] combines the kernel matrix learning and the spectral clustering to improve clustering performance. The subspace learning [28–30] tries to find a shared latent representation for all views for constructing a similarity matrix and then perform spectral clustering method to obtain clustering results. For example, *Wang et al.* in [31] incorporate the local manifold regularization into concept factorization to drive a common representation for multiple views. The authors in [32] propose two methods to perform multi-view clustering based on low-rank representation and sparse subspace learning. While spectral learning [33–35] is to fuse low-dimensional embedding representations from multiple views, and then perform K-means method on the fused embedding representation to generate the final clusters. Generally, co-training based approaches depend on the conditional independence of multiple views. The differences among multiple views are ignored. Our proposed BIGMC method deals with the differences by automatically learning the weight for each view. Subspace based algorithms are sensitive to the quality of original feature representations. That is to say, they cannot find the underlying representation of the data with outliers. Our method constructs the induced sub-graph of the neighboring anchors for each data point and thus reduces the negative effect of outliers for the entire dataset. Additionally, multi-kernel based methods are sensitive to the selection of base kernels. Our method uses an effective initialization method to initialize anchors, and the experiments in Section 6.2.1 demonstrate its robustness. Most spectral based methods need an additional clustering step to generate the final clusters. Our method can obtain them directly without the additional clustering step.

The existing graph-based multi-view clustering methods are related to the above-mentioned multi-view spectral clustering methods [36, 37]. The difference is that the former method forms clusters on the unified graph of multiple views not on the embedding representation [38]. For most graph-based multi-view clustering methods, they still cannot simultaneously address the limitations mentioned in the introduction. For example, the authors in [37] utilize a two-state learning strategy, where they first construct the initial graph of each view and then optimize as well as integrate them into a global graph. Both [33] and [36] propose to learn a common graph directly without considering the discriminative information contained in different views. A graph-based multi-view clustering method [16] is proposed to jointly learn multiple view graphs and a fusion graph. It does not take the consensus information into account and also has a high computational complexity. To this end, two multi-view spectral clustering methods via bipartite graph are presented [17, 39]. While [17] keeps the selected salient points fixed and thus is sensitive to the initialization. Both [17] and [39] construct the Laplacian matrix for each view and keep them fixed during fusion. Additionally, K-means is required to obtain the final clusters. Our proposed BIGMC can alleviate all these limitations. In the experiment section, some representative methods will be compared to our method.

3 Proposed Method

Before presenting our proposed BIGMC method, we first introduce some notations. Throughout the paper, for a matrix $X \in \mathbb{R}^{d \times n}$, let x_j be the j-th column vector, x_{ij} be the (i, j)-th entry, $Tr(X)$ be the trace, and $||X||_F$ be the Frobenius norm respectively. For a vector $\mathrm{x} \in \mathbb{R}^{d \times 1}$, we denote x_j as j-th entry, x^T as the transpose, and $||\mathrm{x}||_p = (\sum_{i=1}^{d} |x_i|^p)^{1/p}$ as l_p-norm. The identity matrix can be denoted by $\mathbf{I}$, and a vector with all entries of one can be denoted by $\mathbf{1}$.

The bipartite graph can be learned based on the similarities between data points and their corresponding neighbor anchor points [17]. For a multi-view data set with m views, we denote $X^1, \ldots, X^m$ as the data matrices and $X^v = [\mathrm{x}_1^v, \ldots, \mathrm{x}_n^v] \in \mathbb{R}^{d_v \times n}$ as the v-th view data with d_v dimensions as well as n data points. For X^v, let x_j^v be the j-th column vector and x_{ij}^v be the (i, j)-th entry. Let $A^1, \ldots, A^m$ be the uniform anchor matrices and $A^v = [\mathrm{a}_1^v, \ldots, \mathrm{a}_t^v] \in \mathbb{R}^{d_v \times t}$ as the anchor matrix of X^v with d_v dimensions as well as t anchor points. c is the required number of clusters. It is noteworthy that all view data have t consensus anchor points, where each anchor point is the centroid of the corresponding sub-cluster. When $t = c$, each cluster only has one anchor point. When $c < t < n$, each cluster can be represented by several sub-clusters and thus has several anchor points. The specific number of the anchor points for each cluster can be learned by our proposed BIGMC method.

3.1 View Graph Learning

The similarity matrices between data and anchors can be denoted as $S^1, \ldots, S^m$, where $S^v \in \mathbb{R}^{n \times t}$. For the i-th data point x_i^v of X^v, we can connect the j-th anchor a_j^v to it as a neighboring anchor with the probability s_{ij}^v. In general, closer x_i^v and a_j^v are likely to have larger connection probability s_{ij}^v [16]. Thus s_{ij}^v is inversely proportional to the distance between them, e.g. $||\mathrm{x}_i^v - \mathrm{a}_j^v||$. Therefore, when $\{A^v\}_{v=1}^m$ are fixed, the graphs for all views can be learned as follow:

$$\begin{aligned} &\min_{\{S^v\}_{v=1}^m} \sum_{v=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{t} ||\mathrm{x}_i^v - \mathrm{a}_j^v||_2^2 s_{ij}^v + \alpha \sum_{v=1}^{m} ||S^v||_F^2 \\ &s.t.\ \forall v,\ s_{ij}^v \geq 0,\ \mathbf{1}^T \mathrm{s}_i^v = 1. \end{aligned} \tag{1}$$

where the second term is a regularization term, and the parameter α is employed to control the connection sparsity between data points and multiple anchors. If $\alpha = 0$, there is a trivial solution for problem (1), i.e., $s_{ij}^v = 1$ indicating that only its nearest anchor a_j^v can be connected to x_i^v. This is called hard partition. If α is large enough, the connections from all t anchors $\{\mathrm{a}_j^v\}_{j=1}^t$ to x_i^v can be built with the same probability $1/t$. The value of α can be determined adaptively as shown in Section 4.1. The normalization $\mathbf{1}^T \mathrm{s}_i^v = 1$ can be considered as the sparse constraint on S^v.

Here, we learn the view graphs independently via constructing a similarity matrix for each view when fixing the anchor set. The reason is that each graph is only related to each other by the anchor set. Then, we produce a unified bipartite graph matrix and use it to update $\{A^v\}_{v=1}^m$ adaptively until convergence.

3.2 Unified Graph Learning

As mentioned above, our proposed BIGMC can jointly learn the graphs of all views, construct a unified bipartite graph, and automatically determine the importance of each view. To be specific, the unified bipartite graph can be obtained through a unified matrix $U \in \mathbb{R}^{n\times t}$ from $\{S^v\}_{v=1}^m$. Thus we have the following problem:

$$\begin{aligned} &\min_U \sum_{v=1}^{m} ||U - S^v||_F^2 \, \delta_v \\ &s.t.\ \forall i,\ u_{ij} \geq 0,\ \mathbf{1}^T \mathrm{u}_i = 1. \end{aligned} \tag{2}$$

where δ_v represents the weight of v-th view, $\mathrm{u}_i \in \mathbb{R}^{t\times 1}$ is a column vector of U, and u_{ij} is the j-th entry of u_i. The values of the weights $\delta = \{\delta_1, \ldots, \delta_m\}$ can be determined automatically according to Theorem 1 [16] as follows:

Theorem 1. *The weight δ_v can be determined by* $\delta_v = \dfrac{1}{2\sqrt{||U - S^v||_F^2}}$.

Proof. An auxiliary function is defined as follows:

$$\begin{aligned} &\min_U \sum_{v=1}^{m} ||U - S^v||_F \\ &s.t.\ \forall i,\ u_{ij} \geq 0,\ \mathbf{1}^T \mathrm{u}_i = 1 \end{aligned} \tag{3}$$

The Lagrange function of Problem (3) can be written as:

$$\sum_{v=1}^{m} ||U - S^v||_F + \Theta(\Lambda, U) \tag{4}$$

where $\Theta(\Lambda, U)$ is the formalized term derived from the constraints in problem (3), and Λ is the Lagrange multiplier. Then we take the derivative of Problem (4) with respect to U and set it to zero.

$$\sum_{v=1}^{m} (\widehat{\delta_v}) \frac{\partial ||U - S^v||_F^2}{\partial U} + \frac{\partial \Theta(\Lambda, U)}{\partial U} = 0 \tag{5}$$

where $\widehat{\delta_v}$ is given by

$$\widehat{\delta_v} = \frac{1}{2\sqrt{||U - S^v||_F^2}}. \tag{6}$$

Then we obtain the Lagrange function of Eq. (2) as follows:

$$\sum_{v=1}^{m} (\delta_v) \frac{\partial ||U - S^v||_F^2}{\partial U} + \frac{\partial \Theta(\Lambda, U)}{\partial U} = 0 \tag{7}$$

From Eq. (5) and (7), we can get the same solution to Eq. (3) and Eq. (2) if $\delta_v = \widehat{\delta_v}$. In this case, the solution to the weight δ_v is

$$\delta_v = \frac{1}{2\sqrt{||U - S^v||_F^2}}. \tag{8}$$

Eq. (5) cannot be solved directly since δ_v depends on the target variable U when S^v is given. However, if δ_v is set stationary, Eq. (5) can be regarded as the solution to Eq. (2). In this case, the calculated U from Eq. (5) (it is in fact Eq. (22) shown below) will be further employed to update δ_v via Eq. (8). This strategy inspires us to solve the Problem (3) through an iterative way. Moreover, if the iterative optimization strategy converges (shown in Section 4), the converged values of U and S^v are optimal. Similarly, the weight δ_v is correspondingly tuned to an optimal value by Eq. (8). Hence, Problem (2) can be transformed into problem (3) when the weights δ are determined by Eq. (8), where the values of U and S^v are obtained in the last iteration.

Problem (1) and Problem (2) can be combined to learn $\{S^v\}_{v=1}^m$ and U jointly as follows:

$$\begin{aligned} &\min_{\{S^v\}_{v=1}^m, U} \sum_{v=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{t} ||\mathrm{x}_i^v - \mathrm{a}_j^v||_2^2 s_{ij}^v + \alpha \sum_{v=1}^{m} ||S^v||_F^2 \\ &+ \sum_{v=1}^{m} ||U - S^v||_F^2 \, \delta_v \\ &s.t.\ \forall v, i,\ s_{ij}^v \geq 0,\ \mathbf{1}^T \mathrm{s}_i^v = 1,\ u_{ij} \geq 0,\ \mathbf{1}^T \mathrm{u}_i = 1. \end{aligned} \tag{9}$$

We notice that the matrices $\{S^v\}_{v=1}^m$ and U can be learned jointly in a problem when $\{A^v\}_{v=1}^m$ are fixed. In the next subsection, we can adaptively find the consensus anchor points of all views.

3.3 Consensus Anchor Learning

When the unified matrix U is updated, we can explore the consensus anchors and reposition them in all views. For j-th subcluster of v-th view data, its anchor a_j^v can be obtained based on the mean of all data points connected to it by

$$\mathrm{a}_j^v = \frac{\sum_{i=1}^n u_{ij} \mathrm{x}_i^v}{\sum_{i=1}^n u_{ij}} \tag{10}$$

where $\mathrm{a}_j^v \in \mathbb{R}^{d_v \times 1}$ and $j = 1, \ldots, t$. Then the anchor matrices $\{A^v\}_{v=1}^m$ can be updated. At last, we combine Eq. (10) with problem (9) and learn the matrices $\{S^v\}_{v=1}^m$, U, and $\{A^v\}_{v=1}^m$ jointly such that they can assist each other in the iteration process.

$$\begin{aligned} &\min_{\{S^v\}_{v=1}^m, U, \{A^v\}_{v=1}^m} \sum_{v=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{t} ||\mathrm{x}_i^v - \mathrm{a}_j^v||_2^2 s_{ij}^v + \alpha \sum_{v=1}^{m} ||S^v||_F^2 \\ &+ \sum_{v=1}^{m} ||U - S^v||_F^2 \, \delta_v \\ &s.t.\ \forall v, i,\ s_{ij}^v \geq 0,\ \mathbf{1}^T \mathrm{s}_i^v = 1,\ u_{ij} \geq 0,\ \mathbf{1}^T \mathrm{u}_i = 1. \end{aligned} \tag{11}$$

3.4 Optimal Bipartite Graph Learning

As mentioned above, the edge weights of the bipartite graph can be represented by $U \in \mathbb{R}^{n\times t}$, where each element, u_{ij}, is a weight of the edge that connects x_i and the corresponding a_j of all views [40]. In this case, the weighted adjacency matrix, $Z \in \mathbb{R}^{(n+t)\times(n+t)}$, and the degree matrix, D_U, can have the following block structures:

$$Z = \begin{bmatrix} 0 & U \\ U^T & 0 \end{bmatrix},\ D_U = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$$

where $D_1 \in \mathbb{R}^{n\times n}$; i-th vector of D_1 is $\mathrm{d}_i^1 = \sum_{j=1}^t u_{ij}$; $D_2 \in \mathbb{R}^{t\times t}$; j-th vector of D_2 is $\mathrm{d}_j^2 = \sum_{i=1}^n u_{ij}$. Hence, the normalized Laplacian matrix is given by

$$L_U = \mathbf{I} - (D_U)^{-1/2} Z (D_U)^{-1/2}. \tag{12}$$

The neighbor anchor assignment is optimal for each data point in all views when there exist exactly c connected components in the bipartite graph. It can be achieved by imposing a rank constraint on L_U of the bipartite graph Z associated with U. As pointed out by [41], first, the eigenvalues of L_U are in a normalized form that enables the spectra relate better to graph invariants for general graphs to some extent than a standard form. Second, there is an important property of D_U if U is non-negative:

Theorem 2. *The multiplicity c of the eigenvalue 0 of the normalized Laplacian matrix L_U equals the number of connected components in the bipartite graph associated with U.*

The proof of Theorem 2 has been shown in [41]. The Theorem 1 says that the n data points and t anchors can be partitioned into c clusters based on Z related to U if $rank(L_U) = (n + t) - c$. Hence, the final subclusters and clusters can be generated without need to perform an additional clustering method. The optimal bipartite graph can be learned by solving the following problem:

$$\begin{aligned} \min_{\{S^v\}_{v=1}^m, U, \{A^v\}_{v=1}^m} & \sum_{v=1}^m \sum_{i=1}^n \sum_{j=1}^t ||\mathrm{x}_i^v - \mathrm{a}_j^v||_2^2 s_{ij}^v + \alpha \sum_{v=1}^m ||S^v||_F^2 \\ & + \sum_{v=1}^m ||U - S^v||_F^2 \, \delta_v \qquad (13) \\ & s.t. \ \forall v, i, \ s_{ij}^v \geq 0, \ \mathbf{1}^T \mathrm{s}_i^v = 1, \ u_{ij} \geq 0, \ \mathbf{1}^T \mathrm{u}_i = 1, \\ & rank(L_U) = (n + t) - c. \end{aligned}$$

It can be noticed that the constraint $rank(L_U) = (n + t) - c$ is nonlinear and hard to solve. To relax this constraint, we introduce c-smallest eigenvalues of L_U, denoted by $\{\eta_q(L_U)\}_{q=1}^c$, in which $\eta_q(L_U) \geq 0$ since L_U is positive semi-definite. Thus let $\sum_{q=1}^c \eta_q(L_U) = 0$ so that the rank constraint can be achieved. From Ky Fan's Theorem [42], this problem can be turned into

$$\sum_{q=1}^c \eta_q(L_U) = \min_{F \in \mathbb{R}^{(n+t)\times c}, F^T F = \mathbf{I}} Tr(F^T L_U F). \qquad (14)$$

Therefore, we can obtain the objective function by plugging problem (14) into problem (13).

$$\begin{aligned} \min_{\{S^v\}_{v=1}^m, U, \{A^v\}_{v=1}^m, F} & \sum_{v=1}^m \sum_{i=1}^n \sum_{j=1}^t ||\mathrm{x}_i^v - \mathrm{a}_j^v||_2^2 s_{ij}^v + \alpha \sum_{v=1}^m ||S^v||_F^2 \\ & + \sum_{v=1}^m ||U - S^v||_F^2 \, \delta_v + \beta Tr(F^T L_U F) \qquad (15) \\ & s.t. \ \forall v, i, \ s_{ij}^v \geq 0, \ \mathbf{1}^T \mathrm{s}_i^v = 1, \ u_{ij} \geq 0, \ \mathbf{1}^T \mathrm{u}_i = 1, F^T F = \mathbf{I}. \end{aligned}$$

When the parameter β is large enough, the optimal U obtained by solving problem (15) can make $\sum_{q=1}^c \eta_q(L_U) = 0$ achieved. Note that we can use β to control the number of connected components in the bipartite graph, denoted by γ. β will be increased when $\gamma < c$ and decreased when $\gamma > c$ in each iteration. Hence, the resulting bipartite graph matrix Z has exact c connected components, and groups n data points as well as t anchors into c clusters. We can solve the problem (15) by an alternating optimization strategy.

4 Optimization Strategy

It has been a challenging issue that each variable in problem (15) can have an optimized solution since they are coupled together. An alternating iterative strategy [17] can effectively transform a constrained optimization problem into a series of unconstrained sub-problems by plugging some penalty terms into the objective function. In this paper, we have variables $\{S\}_{v=1}^m$, $\{\delta_v\}_{v=1}^m$, U, F, and $\{A_v\}_{v=1}^m$ to be optimized. The strategy is that one of them is updated when the others are fixed. Specifically, the updated rules are presented in the subsections.

4.1 Fix $\{\delta_v\}_{v=1}^m$, U, F, and $\{A_v\}_{v=1}^m$, Update $\{S\}_{v=1}^m$

When we fix $\{\delta_v\}_{v=1}^m$, U, F, and $\{A_v\}_{v=1}^m$ of the problem (15), which makes the last term a constant. In this case, the problem becomes:

$$\begin{aligned} \min_{\{S^v\}_{v=1}^m} & \sum_{v=1}^m \sum_{i=1}^n \sum_{j=1}^t ||\mathrm{x}_i^v - \mathrm{a}_j^v||_2^2 s_{ij}^v + \alpha \sum_{v=1}^m ||S^v||_F^2 \\ & + \sum_{v=1}^m ||U - S^v||_F^2 \, \delta_v \qquad (16) \\ & s.t. \ \forall v, i, \ s_{ij}^v \geq 0, \ \mathbf{1}^T \mathrm{s}_i^v = 1. \end{aligned}$$

It is easy to be noticed that the updates of $\{S\}_{v=1}^m$ are independent for all views and not coupled together. Thus, S^v can be updated individually by the following problem:

$$\begin{aligned} \min_{S^v} & \sum_{i=1}^n \sum_{j=1}^t ||\mathrm{x}_i^v - \mathrm{a}_j^v||_2^2 s_{ij}^v + \alpha ||S^v||_F^2 \\ & + ||U - S^v||_F^2 \, \delta_v \qquad (17) \\ & s.t. \ \forall i, \ s_{ij}^v \geq 0, \ \mathbf{1}^T \mathrm{s}_i^v = 1. \end{aligned}$$

Besides, we can also find that updating s_i^v for each vector is independent and get s_i^v by optimizing the following function:

$$\begin{aligned} \min_{\mathrm{s}_i^v} & \sum_{j=1}^t ||\mathrm{x}_i^v - \mathrm{a}_j^v||_2^2 s_{ij}^v + \alpha ||\mathrm{s}_i^v||_2^2 \\ & + ||\mathrm{u}_i - \mathrm{s}_i^v||_2^2 \, \delta_v \qquad (18) \\ & s.t. \ \forall i, \ s_{ij}^v \geq 0, \ \mathbf{1}^T \mathrm{s}_i^v = 1. \end{aligned}$$

For the convenience of calculation, let θ_i as a vector with j-th element $\theta_{ij} = ||\mathrm{x}_i^v - \mathrm{a}_j^v||$. Then the problem (18) can be rewritten as

$$\begin{aligned} \min_{\mathrm{s}_i^v} & \frac{1}{2} ||\mathrm{s}_i^v + \frac{\theta_i}{2\alpha}||_2^2 + \frac{1}{2\alpha} ||\mathrm{u}_i - \mathrm{s}_i^v||_2^2 \, \delta_v \qquad (19) \\ & s.t. \ \forall i, \ s_{ij}^v \geq 0, \ \mathbf{1}^T \mathrm{s}_i^v = 1. \end{aligned}$$

The problem (19) can be tackled with a closed form if we constrain s_i^v having k nonzero elements. That is to say, only k-nearest anchors for each data point x_i^v are taken into account instead of k-nearest data points. This assignment of multiple neighboring anchors contributes to preserving both the invariant and discriminative local structures since each object in different views has not only invariances but also discrepancies. From [16], we have

$$\alpha = \frac{1}{2}(k\theta_{i,k+1} - \sum_{a=1}^k \theta_{ia} - 2k\delta_v u_{i,k+1} - 2\delta_v) \qquad (20)$$

and the final optimized solution of s_{ij}^v

$$\mathrm{s}_{ij}^v = \begin{cases} \dfrac{\theta_{i,k+1} - \theta_{ij} + 2\delta_v(u_{ij} - u_{i,k+1})}{k\theta_{i,k+1} - \sum_{a=1}^k \theta_{ia} - 2k\delta_v u_{i,k+1} + 2\sum_{a=1}^k \delta_v u_{ia}} & j \leq k \\ 0 & j > k \end{cases} \qquad (21)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3021649, IEEE Transactions on Knowledge and Data Engineering

4.2 Fix $\{S\}_{v=1}^{m}$, U, F, and $\{A_v\}_{v=1}^{m}$, Update $\{\delta_v\}_{v=1}^{m}$

When we fix $\{S\}_{v=1}^{m}$, U, F, and $\{A_v\}_{v=1}^{m}$, solving problem (15) to update $\{\delta_v\}_{v=1}^{m}$ can be turned into solving problem (2). As mentioned above, the final solution of each δ_v in $\{\delta_v\}_{v=1}^{m}$ can be obtained according to Eq. (8).

4.3 Fix $\{S\}_{v=1}^{m}$, $\{\delta_v\}_{v=1}^{m}$, F, and $\{A_v\}_{v=1}^{m}$, Update U

When we fix $\{S\}_{v=1}^{m}$, $\{\delta_v\}_{v=1}^{m}$, F, and $\{A_v\}_{v=1}^{m}$, the problem (15) can be transformed into

$$\begin{aligned} &\min_{U} \sum_{v=1}^{m} ||U - S^v||_F^2 \,\delta_v + \beta Tr(F^T L_U F) \\ &s.t.\ \ \forall i,\ u_{ij} \geq 0,\ \mathbf{1}^T \mathrm{u}_i = 1. \end{aligned} \tag{22}$$

where all L_U, D_U, and Z depend on U. Specifically, the last term reveals the mutual relations as follows:

$$\begin{aligned} Tr(F^T L_U F) &= \frac{1}{2} \sum_{i=1}^{n+t} \sum_{j=1}^{n+t} z_{ij} || \frac{\mathrm{f}_i}{\mathrm{d}_i^1} - \frac{\mathrm{f}_j}{\mathrm{d}_j^2} ||_2^2 \\ &= \sum_{i=1}^{n} \sum_{j=1}^{t} u_{ij} \mu_{ij} \end{aligned} \tag{23}$$

where $\mu_{ij} = ||\frac{\mathrm{f}_i}{\mathrm{d}_i^1} - \frac{\mathrm{f}_j}{\mathrm{d}_j^2}||_2^2$. Then, the problem (22) can be rewritten as:

$$\begin{aligned} &\min_{U} \sum_{v=1}^{m} ||U - S^v||_F^2 \,\delta_v + \beta \sum_{i=1}^{n} \sum_{j=1}^{t} u_{ij} \mu_{ij} \\ &s.t.\ \ \forall i,\ u_{ij} \geq 0,\ \mathbf{1}^T \mathrm{u}_i = 1. \end{aligned} \tag{24}$$

Similarly, updating u_i for each row in U is independent. We can have

$$\begin{aligned} &\min_{\mathrm{u}_i} \sum_{v=1}^{m} ||\mathrm{u}_i - \mathrm{s}_i^v||_2^2 \,\delta_v + \beta \mu_i^T \mathrm{u}_i \\ &s.t.\ \ \forall i,\ u_{ij} \geq 0,\ \mathbf{1}^T \mathrm{u}_i = 1. \end{aligned} \tag{25}$$

We denote ϕ and φ as the Lagrange multipliers for the two constraints. Thus we can have the Lagrange function of problem (25):

$$\begin{aligned} \mathcal{L}(\mathrm{u}_i, \phi, \varphi) = &\sum_{v=1}^{m} ||\mathrm{u}_i - \mathrm{s}_i^v||_2^2 \,\delta_v + \beta \mu_i^T \mathrm{u}_i \\ &- \phi(\mathbf{1}^T \mathrm{u}_i - 1) - \varphi^T \mathrm{u}_i. \end{aligned} \tag{26}$$

Then, we take the derivative of $\mathcal{L}$ with respect to u_i, set it to zero, and obtain the following equation:

$$2\mathrm{u}_i \sum_{v=1}^{m} \delta_v - 2 \sum_{v=1}^{m} \mathrm{s}_i^v \delta_v + \beta \mu_i - \phi \mathbf{1} - \varphi = \mathbf{0} \tag{27}$$

From [43], only the optimal u_i^*, ϕ^*, and φ^* can satisfy the Eq. (27). Additionally, β can be adaptively determined and thus be treated as a known parameter. Let $a = 2\sum_{v=1}^{m} \delta_v$ and $\mathrm{p}_i = 2\sum_{v=1}^{m} \mathrm{s}_i^v \delta_v - \beta \mu_i$ for the constants. On the basis of the Karush-Kuhn-Tucker (KKT) conditions, we can have

$$\mathrm{u}_i^* a - \mathrm{p}_i - \phi^* \mathbf{1} - \varphi^* = \mathbf{0} \tag{28}$$

where $\forall j, \mathrm{u}_{ij}^* \geq 0, \varphi_j^* \geq 0, \mathrm{u}_{ij}^* \varphi_j^* = 0$. According to the constraint $\mathbf{1}^T \mathrm{u}_i = 1$, i.e., $\mathbf{1}^T \mathrm{u}_i^* = 1$ we can obtain

$$\phi^* = \frac{a - \mathbf{1}^T \mathrm{p}_i - \mathbf{1}^T \varphi^*}{t} \tag{29}$$

Plugging ϕ^* into Eq. (28), we can have

$$\begin{aligned} \mathrm{u}_i^* &= \frac{\mathrm{p}_i}{a} + \frac{\mathbf{1}}{t} - \frac{\mathbf{1}^T \mathrm{p}_i \mathbf{1}}{at} - \frac{\mathbf{1}^T \varphi^* \mathbf{1}}{at} + \frac{\varphi^*}{a} \\ &= (\mathrm{w}_i - \sigma^* \mathbf{1})_+ \end{aligned} \tag{30}$$

where $\mathrm{w}_i = \frac{\mathrm{p}_i}{a} + \frac{\mathbf{1}}{t} - \frac{\mathbf{1}^T \mathrm{p}_i \mathbf{1}}{at}$, $\sigma^* = \frac{\mathbf{1}^T \varphi^*}{at}$, $\frac{\varphi^*}{a} \geq 0$, $(\cdot)_+ = \max(\cdot, 0)$, and $u_{ij}^* = (w_{ij} - \sigma^*)_+$. We can also derive $\varphi_j^* = a(\sigma^* - w_{ij} + u_{ij}^*) = a(\sigma^* - w_{ij})_+$. Thus $\sigma^* = \frac{\sum_{j=1}^{t}(\sigma^* - w_{ij})_+}{t}$. The solution σ^* can be obtained by finding the root of problem as

$$f(\sigma) = \sigma - \frac{\sum_{j=1}^{t}(\sigma - w_{ij})_+}{t}. \tag{31}$$

where $\sigma \geq 0$ and $f'(\sigma) \geq 0$. It can be noted that $f(\sigma)$ is a linear and convex function. The Newton-Raphson method as a root-finding algorithm can generate a successively approximation to the root of a real-valued function. Hence, a sufficiently precise value of σ (i.e., σ^*) is reached by iterating computing a better approximation, $\sigma_{\tau+1}$, to the root. Solving for $\sigma_{\tau+1}$ gives

$$\sigma_{\tau+1} = \sigma_\tau - \frac{f(\sigma)}{f'(\sigma)} \tag{32}$$

Therefore, u_i^* can be obtained by Eq. (30) for each row of U.

4.4 Fix $\{S\}_{v=1}^{m}$, $\{\delta_v\}_{v=1}^{m}$, U, and $\{A_v\}_{v=1}^{m}$, Update F

When we fix $\{S\}_{v=1}^{m}$, $\{\delta_v\}_{v=1}^{m}$, U, and $\{A_v\}_{v=1}^{m}$, F can be updated by handling the following problem:

$$\min_{F} Tr(F^T L_U F)\ s.t.\ F^T F = \mathbf{I} \tag{33}$$

We can rewrite F as the block matrix

$$F = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

where $F_1 \in \mathbb{R}^{n \times c}$ and $F_2 \in \mathbb{R}^{t \times c}$. According to Eq. (12), the problem (33) can be rewritten as

$$\begin{aligned} &\max_{F^T F = \mathbf{I}} Tr(F^T (D_U)^{-1/2} Z (D_U)^{-1/2} F) \\ &\Rightarrow \max_{F_1^T F_1 + F_2^T F_2 = \mathbf{I}} Tr(F_1^T (D_1)^{-1/2} U (D_2)^{-1/2} F_2) \end{aligned} \tag{34}$$

The problem (34) can be solved by Lemma 1 [40, 44]. In Lemma 1, $B = (D_1)^{-1/2} U (D_2)^{-1/2}$.

Lemma 1. *Given $F_1 \in \mathbb{R}^{n \times c}$, $B \in \mathbb{R}^{n \times t}$, and $F_2 \in \mathbb{R}^{t \times c}$. The optimal solutions to the problem*

$$\max_{F_1^T F_1 + F_2^T F_2 = \boldsymbol{I}} Tr(F_1^T B F_2)$$

are $F_1 = \frac{\sqrt{2}}{2} B_1$ and $F_2 = \frac{\sqrt{2}}{2} B_2$, in which B_1 are the leading c left singular vectors of B and B_2 are the leading c right singular vectors of B.

The optimal F is composed of the optimal F_1 and F_2.

4.5 Fix $\{S\}_{v=1}^{m}$, $\{\delta_v\}_{v=1}^{m}$, U, and F, Update $\{A_v\}_{v=1}^{m}$

When we fix $\{S\}_{v=1}^{m}$, $\{\delta_v\}_{v=1}^{m}$, U, and F, each a_j^v can be updated by Eq. (10).

The details are shown in Algorithm 1.

Algorithm 1 BIGMC Optimization Method

```
Input: Data set of $m$ views $X^1, \dots, X^m$ with $X^v \in \mathbb{R}^{d_v \times n}$, the
    number of anchor points $t$, the number of clusters $c$, the
    number of anchor neighbors $k$, initial parameter $\beta$.
Output: The final cluster labels $Y$
Initialize $t$ uniform anchor points for each anchor set
    $A^v$ using an initialization method (e.g., $k$-means) on
    concatenate data from all views.
Construct the bipartite graph matrix $S^v$ for each view.
Set the weight for each view $\delta_v = 1/m$.
Construct $U$ based on $\{S^v\}_{v=1}^m$ and $\delta$.
Calculate $F$ by solving problem (32).
repeat
    repeat
        Fix $\delta$, $U$, $F$, and $A$, update $\{S^v\}_{v=1}^m$ by Eq. (21).
        Fix $\{S^v\}_{v=1}^m$, $U$, $F$, and $A$, update $\delta$ by Eq. (8).
        Fix $\{S^v\}_{v=1}^m$, $\delta$, $F$, and $A$, update $U$ by Eq. (30).
        Fix $\{S^v\}_{v=1}^m$, $\delta$, $U$, and $A$, update $F$ by Lemma 1.
    until Theorem 1 or the maximum iteration reached.
    Fix $\{S^v\}_{v=1}^m$, $\delta$, $U$, and $F$, update $A$ by Eq. (10).
until converge
The final clusters $Y$ are the exact $c$ components in the
    unified bipartite graph matrix $U$.
```

5 Complexity and Convergence Analysis

5.1 Complexity Analysis

From Algorithm 1, the computational complexity of our proposed BIGMC method consists mainly of six parts, which correspond the initialization and updates of our variables respectively. To be more specific, the update of $\{S^v\}_{v=1}^m$ takes $O(mnt)$, where m is the number of views; n is the number of data objects; t is the number of anchor points and $c \leq t \ll n$; c is the number of required number of clusters. The update of weights of all views δ has the computational complexity of $O(mnt)$. The update of the unified graph matrix U is achieved by solving Eq. (28) taking $O(cn)$. The learning of F takes $O(cnt)$. Hence, this sub-iteration procedure is $O((2mt + c + ct)n\zeta_1)$, where ζ_1 is the number of iterations. Updating the anchor points A needs to cost $O(mntd)$, where $d = \max(d^1, \dots, d^m)$. Moreover, we initialize the anchors $\{A^v\}_{v=1}^m$ by taking $O(ndt)$ with Var-Part method. The initialization of $\{S^v\}_{v=1}^m$ takes $O(mndt)$.

Overall, the computational complexity of BIGMC takes $O(((2mt + c + ct)\zeta_1 + mtd)n\zeta_2 + ndt(m + 1))$, in which ζ_2 is the number of iterations.

5.2 Convergence Analysis

The overall objective function Eq. (15) is not a joint convex optimization problem of variables. Acquiring a globally optimal solution is still an open problem. The problem (15) is solved using the optimization strategy proposed in Section 4. After alternating optimizing variables, the corresponding each sub-problem is convex and the optimal solution of it is given. Specifically, the convergences of all sub-problems can be shown as follows.

For the update of $\{S\}_{v=1}^m$, the objective function of problem (19) is a convex function. The reason for this conclusion is that its second order derivative with respect to s_i^v is equal to 1. Therefore, it is monotonic decreasing using the optimization strategy.

For the update of weights δ, the objective function of problem (2) is a linear convex problem. A closed-form solution of δ is given in Eq. (8).

For the update of U, we can denote $\widehat{U}$ as the updated U in the augmented Lagrangian iteration process and $\Gamma(U) = \beta Tr(F^T L_U F)$. Thus the following inequality can be derived from problem (23) and (8) with the decrease of function error

$$\sum_{v=1}^{m} \frac{||\widehat{U} - S^v||_F^2}{2||\widehat{U} - S^v||_F} + \Gamma(\widehat{U}) \leq \sum_{v=1}^{m} \frac{||U - S^v||_F^2}{2||U - S^v||_F} + \Gamma(U). \tag{35}$$

According to a lemma from [45], the convergence of problem (22) can be obtained.

Lemma 2. *For any non-zero matrix $A' \in \mathbb{R}^{n\times t}$ and $B' \in \mathbb{R}^{n\times t}$, the following inequality holds:*

$$||A'||_F - \frac{||A'||_F^2}{2||B'||_F^2} \leq ||B'||_F - \frac{||B'||_F^2}{2||B'||_F^2} \tag{36}$$

Let $A' = \sum_{v=1}^m (\widehat{U} - S^v)$ and $B' = \sum_{v=1}^m (U - S^v)$, and we can have

$$\begin{aligned} &|| \sum_{v=1}^{m} (\widehat{U} - S^v) ||_F - \frac{|| \sum_{v=1}^m (\widehat{U} - S^v) ||_F^2}{2|| \sum_{v=1}^m (U - S^v) ||_F^2} \leq \\ &|| \sum_{v=1}^{m} (U - S^v) ||_F - \frac{|| \sum_{v=1}^m (U - S^v) ||_F^2}{2|| \sum_{v=1}^m (U - S^v) ||_F^2} \end{aligned} \tag{37}$$

Therefore,

$$\begin{aligned} &\sum_{v=1}^{m} ||\widehat{U} - S^v||_F - \sum_{v=1}^{m} \frac{||\widehat{U} - S^v||_F^2}{2||U - S^v||_F^2} \leq \\ &\sum_{v=1}^{m} ||U - S^v||_F - \sum_{v=1}^{m} \frac{||U - S^v||_F^2}{2||U - S^v||_F^2} \end{aligned} \tag{38}$$

We sum Eq. (35) and (38) over both sides and get

$$\sum_{v=1}^{m} ||\widehat{U} - S^v||_F + \Gamma(\widehat{U}) \leq \sum_{v=1}^{m} ||U - S^v||_F + \Gamma(U) \tag{39}$$

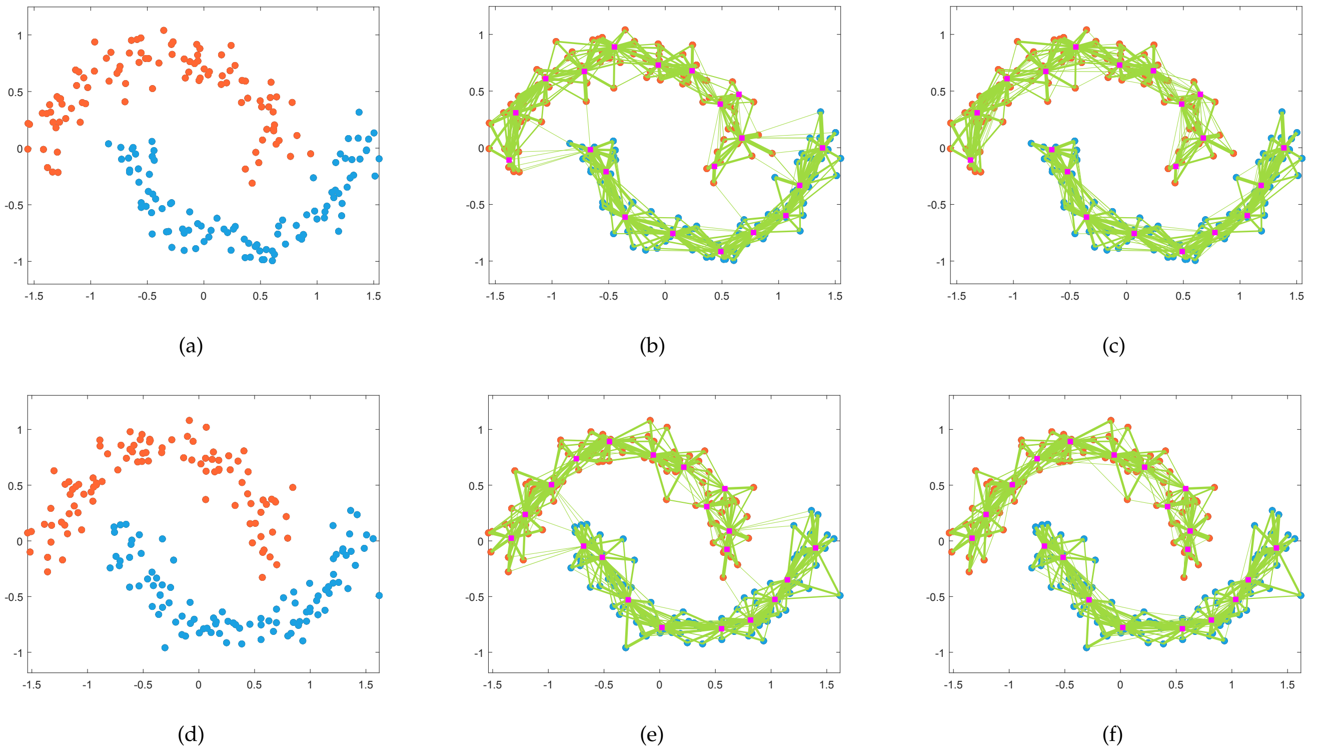
As a result, the convergence of (22) is proved.

For the update of F, the objective function of problem (33) F is updated by Lemma 1 through SVD of B.

For the update of $\{A\}_{v=1}^m$, the problem converges when the connections between data points and anchor points no longer change.

6 Experiments

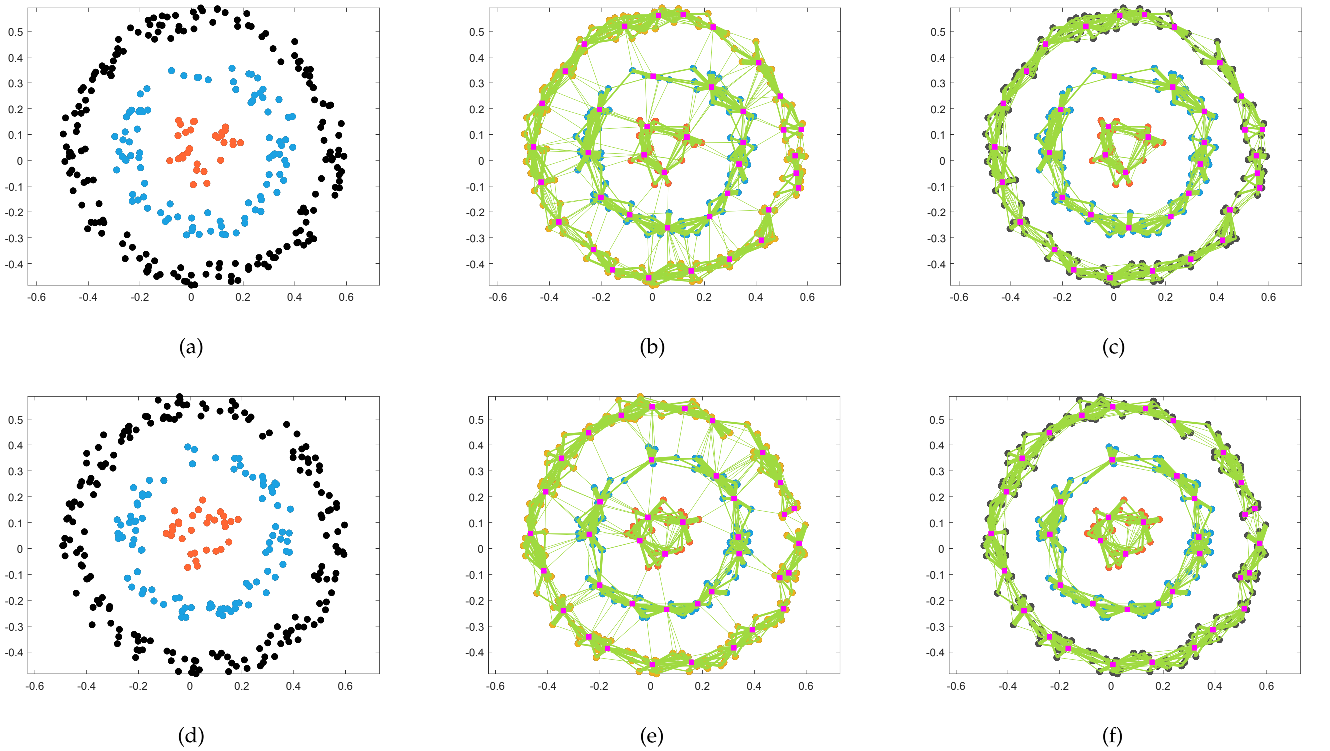
The experiments are conducted on Matlab development environment to compare with the baseline methods. In this section, we investigate the performance of our proposed BIGMC method on both synthetic and real-world data sets. Thus, we present two groups of experiments. The first group is to show the effectiveness of BIGMC by observing the visual illustration of its capability on the synthetic data sets. The learned connections within each view will be shown to prove the capability of similarity learning, where a large line weight indicates a strong connection between the data point and its neighbor anchor point. The second



(a) (b) (c)

(d) (e) (f)

Fig. 2. Clustering results on Two-Moon data set. The upper row includes the original first view data, the learned graph with the learned S^1, and the learned graph with the learned U. The lower row contains the original second view data, the learned graph with the learned S^2, and the learned graph with the learned U. The red dots are cluster 1, and the blue dots are cluster 2. The pink squares are the learned anchor points, and the green lines are the learned connections between data points and anchor points.



(a) (b) (c)

(d) (e) (f)

Fig. 3. Clustering results on Three-Circle data set. The upper row includes the original first view data, the learned graph with the learned S^1, and the learned graph with the learned U. The lower row contains the original second view data, the learned graph with the learned S^2, and the learned graph with the learned U. The red, blue, and black dots are cluster 1, cluster 2, and cluster 3. The pink squares are the learned anchor points, and the green lines are the learned connections between data points and anchor points.

group can be divided into five sub-groups. The first sub-group is to determine an efficient initialization method for BIGMC to initialize the anchor points such that the sensitivity to the initialization can be alleviated. The second sub-group contains the clustering results on real-world datasets to demonstrate the superiority of BIGMC compared with the baselines. Moreover, the learning results of anchors by BIGMC on 3 real-world datasets are presented to show the adaptive ability of learning anchors. The third sub-group is to further evaluate BIGMC by generating 4 variants of BIGMC. The fourth sub-group is to show the convergence results of BIGMC. The fifth sub-group contains the results of running time by BIGMC and baselines. In this paper, we assume all views are complete.

6.1 Experiments on Synthetic Datasets

Data sets. We follow [16, 37] to conduct experiments on two synthetic datasets to evaluate the performance of our proposed BIGMC. The first one contains two views, which are shown in Fig. 2 (a) and (d), called "Two-Moon data set". Each view has two clusters, i.e., one moon pattern with red dots and the other moon pattern with blue dots. Each cluster has 100 data points and adds 0.12 percentage of random Gaussian noises. The second also involves two views, which are shown in Fig. 3 (a) and (d), called "Three-Circle data set". Each view has three clusters respectively represented by 30 red dots, 90 blue dots, and 180 black dots in a circle pattern. The same percentage of random Gaussian noises are added.

Results. For Two-Moon data set, we set the number of anchor neighbors $k = 3$ and the number of anchor points $t = 20$. Fig. 2 (b) and (e) show the learned graphs for the two views, in which the pink squares are the learned anchor points, and lines are generated from the learned S^1 and S^2, respectively. The line weight indicates how much similar each data point and its neighbor anchor point. It can be seen that the two clusters are weakly connected together in both views since there is no low-rank constraint on the view graph matrix. Fig. 2 (c) and (f) show the learned graphs for the two views, where the lines are produced from the learned unified matrix U. The final clusters are separated well. The weak connections are cut and the strong connections are strengthened by fusing the complementary information contained in two views. For Three-Circle data set, we set the number of anchor points $k = 3$ and the number of anchor neighbors $t = 40$. Similar to Fig. 2, the learned graphs for the two views are shown in Fig. 3 (b) and (e) with the lines from the learned S^1 and S^2. The three clusters are connected together, where the connections are closer than that in Two-Moon data set. It is harder to separate the clusters correctly for Three-Circle data set. From the results in Fig. 3 (c) and (f), they are separated well based on the unified matrix U by considering the information in two views. Additionally, it is noticed that the number of anchor points in each cluster can be learned adaptively without being specified, and the learned locations of them are well distributed.

6.2 Experiments on Real-world Datasets

To further assess the effectiveness of our proposed BIGMC, we compare BIGMC with several baselines on real-world datasets.

TABLE 1
Statistics of Experimental Data sets

Datasets	n	m	c	d^1	d^2	d^3	d^4	d^5	d^6
3sources	169	3	6	3560	3631	3068	-	-	-
100leaves	1600	3	100	64	64	64	-	-	-
Caltech-7	1474	6	7	48	40	254	1984	512	928
Caltech-20	2386	6	20	48	40	254	1984	512	928
Mfeat	2000	6	10	216	76	64	6	240	47
WebKB	203	3	4	1703	230	230	-	-	-
YaleB	650	3	10	2500	3304	6750	-	-	-

Data sets. The following data sets are widely used in the literature.

1) *3sources*[1]: There are 3 views from BBC, Reuters, and Guardian. Each view has 169 news, which can be grouped into 6 clusters.

2) *100leaves*[2]: There are 3 views, where each view has 1600 data points from each of 100 plant species leaves. Each object can be described by shape descriptor, fine scale margin, and texture histogram in the 3 views, respectively.

3) *Caltech-7*[3]: It is a subset of Caltech-101 data set, consisting of 6 views. Each view has 1474 images, which can be grouped into 7 clusters, i.e., faces, motorbikes, dollar bill, Garfield, stop sign, and windsor chair.

4) *Caltech-20*[4]: It is also a subset of Caltech-101 data set, consisting of the same 6 views with *Caltech-7*. Each view has 2386 images, which can be partitioned into 20 classes.

5) *Mfeat*[5]: It is the Mfeat handwritten digit data set including handwritten digits (0-9) from the UCI repository. There are 6 views. Each view has 2000 samples and each sample can be represented by 6 types of features.

6) *WebKB*[6]: There are 3 views, in which each view has 203 web-pages and 4 classes. Each web-page can be described by the anchor text of the hyper-like, the content of the page, and the title.

7) *YaleB*[7]: It is a subset of the extended Yale-B data set, i.e., the first 10 classes data. There are 3 views, where each view has 650 face images.

The statistical information of all these data sets are shown in Table 1. n is the number of data objects. m is the number of views. c is the required number of clusters. d^v indicates the dimension of features in v-th view.

6.2.1 *Experiments on Initialization of Anchors*

Initialization methods. The initialization of uniform anchor points is essential for our proposed BIGMC method. K-means is sensitive to the initial placement of the anchor points. To address this problem, many initialization methods have been proposed. Considering the clustering effectiveness and computational efficiency [46], we compare

[1] http://mlg.ucd.ie/datasets/3sources.html

[2] https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set

[3] http://www.vision.caltech.edu/Image_Datasets/Caltech101

[4] http://www.vision.caltech.edu/Image_Datasets/Caltech101

[5] http://archive.ics.uci.edu/ml/datasets/Multiple+Features

[6] https://linqs.soe.ucsc.edu/data

[7] http://vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html

TABLE 2
Clustering results (mean±standard-deviation) with metrics (ACC and NMI) by BIGMC with different initialization methods on 7 real-world datasets

Metrics	Init-Methods	3sources	100leaves	Caltech-7	Caltech-20	Mfeat	WebKB	YaleB	Ave
ACC	K-Means	0.775±0.00	0.915±0.01	0.781±0.02	0.596±0.01	0.920±0.00	0.773±0.00	0.520±0.02	0.759±0.01
	greedy K-Means++	0.789±0.01	0.923±0.01	0.735±0.00	0.607±0.00	0.925±0.02	0.782±0.00	0.566±0.00	0.761±0.01
	PCA-Part	0.794±0.00	**0.929±0.01**	0.783±0.00	0.608±0.00	**0.933±0.01**	0.787±0.00	**0.584±0.00**	**0.774±0.00**
	Var-Part	**0.797±0.00**	0.921±0.00	**0.785±0.00**	**0.611±0.00**	0.932±0.01	**0.795±0.00**	0.575±0.01	**0.774±0.00**
NMI	K-Means	0.669±0.00	0.955±0.00	0.670±0.02	0.600±0.01	0.914±0.00	0.495±0.00	0.500±0.01	0.685±0.01
	greedy K-Means++	0.672±0.00	0.961±0.01	0.702±0.00	0.598±0.01	0.908±0.02	0.523±0.00	0.519±0.00	0.698±0.01
	PCA-Part	0.689±0.00	**0.969±0.00**	**0.710±0.00**	0.610±0.00	**0.917±0.01**	0.524±0.00	**0.551±0.00**	**0.710±0.00**
	Var-Part	**0.705±0.00**	0.960±0.01	0.697±0.00	**0.624±0.00**	0.910±0.00	**0.540±0.00**	0.525±0.01	0.709±0.00

four commonly used initialization methods as follow. All the initialization methods work on the concatenation of features from all views.

1) k-means [27]: Randomly selecting t centers in each round and then choosing the center that most reduces the sum-squared-error (SSE). It has the computational complexity $O(ndtl)$.

2) greedy k-means++ [47]: Probabilistically selecting $\log(t)$ centers in each round and then greedily choosing the center that most reduces the SSE. It has the computational complexity $O(ndts)$.

3) PCA-Part [48]: Starting from an initial cluster that contains the entire data set and then obtaining t clusters by repeating the procedure, where it chooses the cluster with the largest SSE and divides it into two sub-clusters by a hyper-plane that contains the center and is orthogonal to the direction of the eigenvector with the largest eigenvalue of the covariance matrix. It has the computational complexity $O(nd^2t)$

4) Var-Part [48]: Approximating PCA-Part by assuming the covariance matrix of the cluster is diagonal. It has computational complexity $O(ndt)$, which is equal to only one iteration of k-means.

For the above initialization methods, $d = d^1 + \cdots + d^v$; l is the number of iterations; t is the number of centers, i.e., that of anchor points; s is the amount of extra sampling. In terms of the computational complexity, Var-Part method has more efficiency on high-dimensional datasets. In the experiments, one of these initialization methods is chosen to initialize the anchor points of our proposed BIGMC method shown at the Step 1 of Algorithm 1. For each real-world data set, we we empirically set $t = n/5$ and $k = 5$. The initial value of parameter β is set to 1. Its value is adaptively tuned in the optimization procedure of the objective function for each data set. Two common metrics are utilized to evaluate the clustering performance: the accuracy (ACC) and the normalized mutual information (NMI). To randomize the experiments, each method is run for 5 times and the means as well as standard deviations of the metrics are reported.

Results. Table 2 shows the clustering results with two metrics by BIGMC with different initialization methods on seven real-world datasets. From the table, PCA-Part and Var-Part initialization methods have comparable performance. Both of them can produce good initial anchors and perform better than k-means as well as greedy k-means++. In terms of the computational complexity, Var-Part method has superiority. This is because PCA-Part method determines the splitting direction by exploring the principal eigenvector of the covariance matrix in $O(d^2)$ time. However, Var-Part method achieves this via obtaining the coordinate axis with the largest variance in $O(d)$ time. Although PCA-Part method is more efficient than Var-Part method in some cases, the latter can scale to high-dimensional datsets with a lower computational complexity. Therefore, we choose Var-Part method as the initialization method of our proposed BIGMC method.

6.2.2 Experiments on Comparisons

Baselines. The following baseline methods are compared with our proposed BIGMC methods.

1) Multi-View Clustering via Concept factorization (MVCC) [31]: Incorporating the local manifold regularization into concept factorization to drive a common representation for multiple views.

2) Pairwise Multi-view Low-Rank Sparse Subspace Clustering (P-MVLRSSC) [32]: Performing multi-view clustering based on low-rank representation and sparse subspace learning between affinity matrices of the pairs of views.

3) Centroid Multi-view Low-Rank Sparse Subspace Clustering (C-MVLRSSC) [32]: Performing multi-view clustering based on low-rank representation and sparse subspace learning between affinity matrices towards a common centroid.

4) Graph-based Multi-view Clustering (GMC) [16]: Constructing the graph of each view based on the pairwise similarity between any two data samples and fusing them to produce a unified matrix. The final clusters can be obtained from the unified matrix.

5) Multi-View Graph Learning (MVGL) [37]: Learning the initial graph of each view, optimizing it with a rank constraint on the Laplacian matrix, and integrating the optimized graphs into a global graph.

6) Multi-view Spectral Clustering (MVSC) [17]: Learning a bipartite graph for each view, combining them using a local manifold fusion method, and running spectral clustering on the fused graph.

7) Multi-view Learning with Adaptive Neighbours (MLAN) [36]: Performing clustering and local structure learning simultaneously and obtaining an optimal graph without

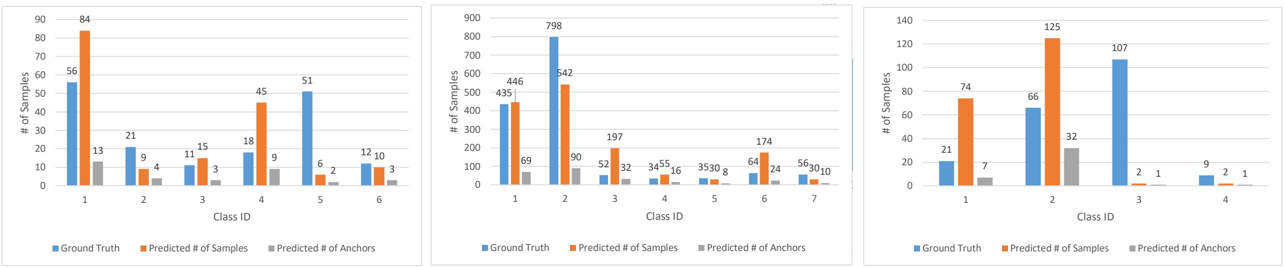
TABLE 3
Clustering results (mean±standard-deviation) with metrics (ACC, NMI, ARI, F-M, PRE, and REC) by different methods on 7 real-world datasets

Metrics	Methods	3sources	100leaves	Caltech-7	Caltech-20	Mfeat	WebKB	YaleB	Ave
ACC	MVCC	0.761±0.01	0.128±0.00	0.471±0.00	0.533±0.00	0.408±0.00	0.709±0.00	0.196±0.00	0.455±0.00
	P-MLRSSC	0.682±0.05	0.030±0.00	0.609±0.08	0.434±0.02	0.592±0.04	0.425±0.03	0.481±0.08	0.465±0.04
	C-MLRSSC	0.662±0.07	0.030±0.00	0.563±0.05	0.429±0.02	0.578±0.05	0.442±0.04	0.478±0.11	0.454±0.05
	GMC	0.692±0.00	0.824±0.00	0.692±0.00	0.456±0.00	0.882±0.00	0.769±0.00	0.434±0.00	0.678±0.00
	MVGL	0.302±0.00	0.766±0.00	0.579±0.00	0.578±0.00	0.856±0.00	0.581±0.00	0.300±0.00	0.566±0.00
	MVSC	0.531±0.00	0.717±0.00	0.621±0.00	0.575±0.00	0.703±0.00	0.567±0.00	0.468±0.00	0.597±0.00
	MLAN	0.763±0.00	0.873±0.01	0.780±0.00	0.525±0.00	**0.973±0.00**	0.729±0.00	0.343±0.00	0.712±0.00
	BIGMC	**0.797±0.00**	**0.921±0.00**	**0.785±0.00**	**0.611±0.00**	0.932±0.01	**0.795±0.00**	**0.575±0.01**	**0.774±0.00**
NMI	MVCC	0.698±0.01	0.552±0.00	0.464±0.00	0.564±0.00	0.422±0.00	0.418±0.00	0.088±0.00	0.458±0.00
	P-MLRSSC	0.594±0.03	0.442±0.01	0.500±0.02	0.487±0.01	0.700±0.02	0.355±0.03	0.378±0.04	0.493±0.02
	C-MLRSSC	0.595±0.03	0.440±0.01	0.497±0.03	0.477±0.01	0.703±0.00	0.376±0.03	0.399±0.02	0.498±0.02
	GMC	0.622±0.00	0.929±0.00	0.660±0.00	0.481±0.00	0.905±0.00	0.435±0.00	0.449±0.00	0.640±0.00
	MVGL	0.109±0.00	0.893±0.00	0.558±0.00	0.576±0.00	0.904±0.00	0.144±0.00	0.271±0.00	0.493±0.00
	MVSC	0.541±0.00	0.886±0.00	0.581±0.00	0.567±0.00	0.831±0.00	0.122±0.00	0.431±0.00	0.565±0.00
	MLAN	0.689±0.00	0.948±0.00	0.636±0.00	0.539±0.00	**0.939±0.00**	0.402±0.00	0.348±0.00	0.643±0.00
	BIGMC	**0.705±0.00**	**0.960±0.01**	**0.697±0.00**	**0.624±0.00**	0.910±0.00	**0.540±0.00**	**0.525±0.01**	**0.709±0.00**
ARI	MVCC	0.631±0.00	0.121±0.00	0.298±0.00	0.487±0.00	0.255±0.00	0.468±0.00	0.028±0.00	0.329±0.00
	P-MLRSSC	0.565±0.06	0.060±0.00	0.324±0.02	0.349±0.05	0.548±0.03	0.246±0.03	0.200±0.02	0.327±0.00
	C-MLRSSC	0.557±0.08	0.059±0.00	0.334±0.03	0.343±0.06	0.559±0.00	0.266±0.03	0.222±0.01	0.334±0.03
	GMC	0.443±0.00	0.497±0.00	0.594±0.00	0.128±0.00	0.850±0.00	0.440±0.00	0.157±0.00	0.444±0.00
	MVGL	-0.036±0.00	0.506±0.00	0.395±0.00	0.263±0.00	0.832±0.00	0.083±0.00	0.093±0.00	0.305±0.00
	MVSC	0.426±0.00	0.318±0.00	0.436±0.00	0.260±0.00	0.694±0.00	0.068±0.00	0.147±0.00	0.336±0.00
	MLAN	0.571±0.00	0.818±0.01	0.572±0.00	0.197±0.01	**0.940±0.00**	0.373±0.00	0.090±0.00	0.509±0.00
	BIGMC	**0.661±0.00**	**0.883±0.01**	**0.690±0.00**	**0.498±0.01**	0.940±0.01	**0.546±0.00**	**0.244±0.02**	**0.615±0.01**
F-M	MVCC	0.734±0.00	0.136±0.00	0.464±0.00	0.541±0.00	0.332±0.00	0.664±0.00	0.148±0.00	0.436±0.00
	P-MLRSSC	0.659±0.05	0.077±0.00	0.518±0.02	0.464±0.04	0.605±0.02	0.445±0.03	0.302±0.02	0.439±0.03
	C-MLRSSC	0.654±0.06	0.076±0.00	0.524±0.02	0.460±0.05	0.615±0.00	0.462±0.03	0.322±0.01	0.444±0.02
	GMC	0.605±0.00	0.504±0.00	0.722±0.00	0340±0.00	0.866±0.00	0.700±0.00	0.265±0.00	0.572±0.00
	MVGL	0.339±0.00	0.513±0.00	0.570±0.00	0.415±0.00	0.850±0.00	0.566±0.00	0.204±0.00	0.494±0.00
	MVSC	0.535±0.00	0.328±0.00	0.647±0.00	0.413±0.00	0.728±0.00	0.564±0.00	0.261±0.00	0.497±0.00
	MLAN	0.683±0.00	0.819±0.01	0.737±0.00	0.371±0.01	0.946±0.00	0.668±0.00	0.211±0.00	0.633±0.00
	BIGMC	**0.751±0.00**	**0.882±0.01**	**0.797±0.00**	**0.557±0.00**	**0.956±0.00**	**0.753±0.00**	**0.350±0.02**	**0.704±0.00**
PRE	MVCC	0.613±0.00	0.076±0.00	0.759±0.00	0.561±0.00	0.322±0.00	0.708±0.00	0.118±0.00	0.461±0.00
	P-MLRSSC	0.707±0.05	0.040±0.00	0.697±0.04	0.426±0.05	0.473±0.03	0.663±0.04	0.234±0.02	0.463±0.03
	C-MLRSSC	0.696±0.05	0.040±0.00	0.711±0.05	0.425±0.05	0.484±0.00	0.682±0.04	0.249±0.01	0.470±0.03
	GMC	0.484±0.00	0.352±0.00	0.886±0.00	0.228±0.00	0.826±0.00	0.592±0.00	0.204±0.00	0.510±0.00
	MVGL	0.218±0.00	0.380±0.00	0.762±0.00	0.327±0.00	0.789±0.00	0.423±0.00	0.164±0.00	0.437±0.00
	MVSC	0.529±0.00	0.205±0.00	0.667±0.00	0.325±0.00	0.651±0.00	0.417±0.00	0.193±0.00	0.427±0.00
	MLAN	0.609±0.00	0.775±0.01	0.739±0.00	0.279±0.00	0.945±0.00	0.559±0.00	0.157±0.00	0.580±0.00
	BIGMC	**0.718±0.00**	**0.870±0.02**	**0.904±0.00**	**0.576±0.01**	**0.953±0.00**	**0.742±0.01**	**0.268±0.01**	**0.688±0.01**
REC	MVCC	0.823±0.00	0.653±0.00	0.334±0.00	0.530±0.00	0.342±0.00	0.626±0.00	0.197±0.00	0.515±0.00
	P-MLRSSC	0.619±0.06	0.771±0.01	0.414±0.03	0.512±0.04	0.843±0.02	0.337±0.03	0.430±0.02	0.561±0.03
	C-MLRSSC	0.619±0.07	0.770±0.02	0.416±0.02	0.503±0.05	0.843±0.00	0.350±0.03	0.455±0.02	0.565±0.03
	GMC	0.805±0.00	0.887±0.00	0.609±0.00	0.673±0.00	0.909±0.00	0.858±0.00	0.378±0.00	0.731±0.00
	MVGL	0.768±0.00	0.789±0.00	0.455±0.00	0.567±0.00	0.920±0.00	0.858±0.00	0.270±0.00	0.647±0.00
	MVSC	0.628±0.00	0.826±0.00	0.629±0.00	0.567±0.00	0.828±0.00	0.873±0.00	0.405±0.00	0.679±0.00
	MLAN	0.777±0.00	0.869±0.00	0.734±0.00	0.557±0.02	0.947±0.00	0.831±0.00	0.321±0.00	0.719±0.00
	BIGMC	**0.834±0.00**	**0.893±0.01**	**0.738±0.00**	**0.698±0.00**	**0.966±0.00**	**0.914±0.01**	**0.495±0.02**	**0.773±0.00**

fusion.

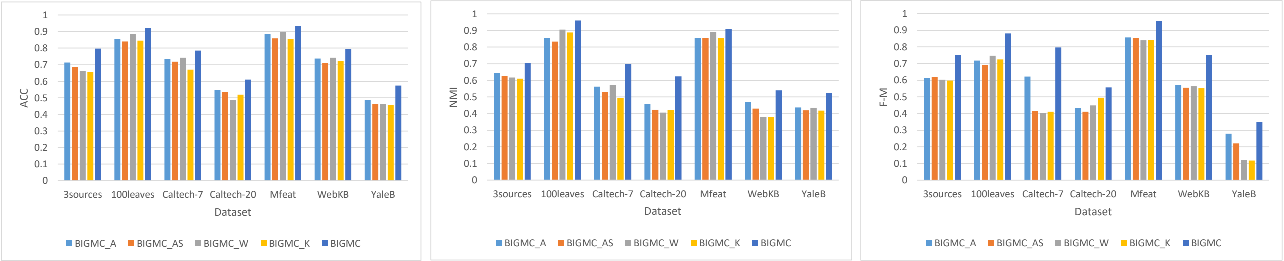
In the baselines, the state-of-the-art comparisons, MVCC, P-MVLRSSC, and C-MVLRSSC methods are based on subspace learning. GMC, MVGL, and MLAN methods are graph-based muti-view clustering. MVSC method is a bipartite graph-based muti-view clustering.

Experiment Settings. For the comparisons, we downloaded the source codes from the authors' websites and followed the experimental setting as well as the parameter tuning steps of their papers. All the baselines and our proposed method are implemented in the Matlab development environment. For BIGMC, we empirically set $t = n/5$ and $k = 5$. The initial value of parameter β is set to 1. Its



(a) (b) (c)

Fig. 4. The learning results of anchors by BIGMC on 3 real-world datasets compared with the ground truth and the predicted number of samples in each class. (a) 3sources; (b) Caltech-7; (c) WebKB.



(a) (b) (c)

Fig. 5. Clustering performance comparison of BIGMC and its 4 variants on 7 real-world datasets in terms of metrics (ACC, NMI, and F-M).

value is adaptively tuned in the optimization procedure of the objective function for each data set. Six common metrics are utilized to evaluate the clustering performance: the accuracy (ACC), the normalized mutual information (NMI), the adjusted rand index (ARI), the F-measure (F-M), the precision (PRE), and the recall (REC). To randomize the experiments, each method is run for 5 times [49, 50] and the means as well as standard deviations of the metrics are reported.

Results. Table 3 shows the clustering results with the six metrics by different methods on the seven real-world datasets. We highlight the best results in bold. From the table, it can be noticed that our proposed BIGMC approach acquires better performance than the baselines.

In terms of ACC, our proposed BIGMC method achieves the best performance for 6 out of 7 datasets. For the Mfeat dataset, BIGMC also finishes the second and performs better than the other methods by a large margin except of MLAN method. In terms of NMI, BIGMC gives a better performance than the comparisons for 6 out of 7 datasets. Moreover, it also achieves the second best performance on Mfeat dataset. We can also see that BIGMC performs better for 6 out of 7 datasets in terms of ARI and has comparable performance with MLAN method. In terms of F-M, PRE, and REC metrics, BIGMC is markedly better than all the baselines on all datasets. To be more specific, BIGMC has a smaller deviation than P-MLRSSC method although they achieve the same average PRE values. Note that the average metric value for each method on all datasets can be seen in the last column. BIGMC, on average, outperforms all the other compared methods.

BIGMC can learn better anchor points based on the learned unified graph. MVCC drives a common consensus representation through manifold regularization and concept factorization. One reason why it is worse than BIGMC is that the constructed Laplacian matrices are fixed during learning process. BIGMC performs better than P-MVLRSSC and C-MVLRSSC, both of which rely on additional K-means clustering method.

Compared with the graph-based methods, i.e., GMC, MVGL, MVSC, and MLAN, BIGMC has a superiority performance since it can learn a better unified graph by learning the individual graph, the unified graph, and the consensus anchor points across all views simultaneously.

To further demonstrate the adaptive ability of learning anchor points, we give an example to show the learning results of anchors by our method on 3 real-world datasets including 3sources, Caltech-7, and WebKB. The ground truth number of samples, the predicted number of samples, and the predicted number of anchors for each class are shown in Fig. 4. From the figure, we can observe that the predicted number of anchors is much smaller than that of samples for each dataset. For different classes, the number of anchors can be learned adaptively, and more anchors can be learned for a class with a larger number of samples.

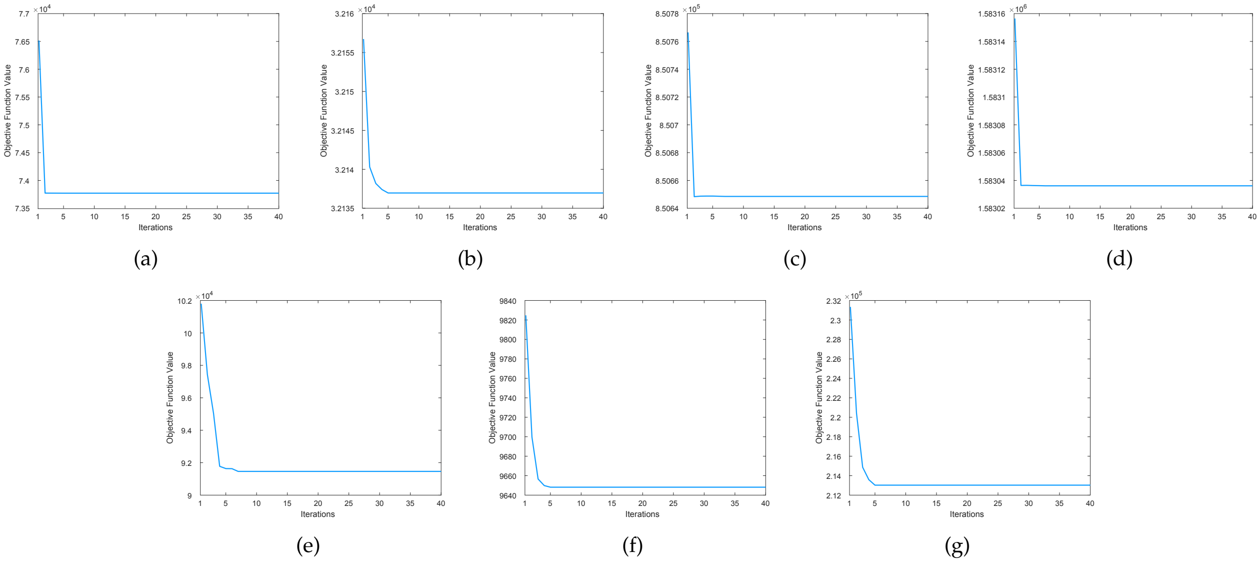
6.2.3 Mode Evaluation

To further show the effectiveness of our proposed BIGMC, we evaluate 4 variants of BIGMC as follow:

1) BIGMC_A: The learned unified graph matrix U are not used to improve the initialized anchors A for each view by removing Steps 6 and 14 in Algorithm 1.

2) BIGMC_AS: The learned unified graph matrix U are not employed to improve both the initialized anchors A and the initialized similarity matrix S for each view by removing Steps 6, 8, and 14 in Algorithm 1.

3) BIGMC_W: The weight of each view δ is set to $1/m$ by removing Step 9 in Algorithm 1.



(a) (b) (c) (d)

(e) (f) (g)

Fig. 6. Convergence curves over different datasets. (a) 3sources. (b) 100leaves. (c) Caltech-7. (d) Caltech-20. (e) Mfeat. (f) WebKB. (g) YaleB.

TABLE 4
Averaged running time by different methods on 7 real-world datasets (in second)

Methods	3sources	100leaves	Caltech-7	Caltech-20	Mfeat	WebKB	YaleB	Ave
MVCC	19.672	107.170	123.957	259.248	167.966	6.354	134.167	116.933
P-MLRSSC	1.016	6.288	155.300	621.127	370.181	0.470	24.193	168.368
C-MLRSSC	0.713	29.177	158.439	679.249	384.128	0.579	25.309	182.513
GMC	0.916	16.143	8.472	21.879	53.744	1.087	1.984	14.889
MVGL	0.634	74.010	169.243	611.509	497.051	0.969	9.092	194.644
MVSC	**0.206**	**4.545**	**3.044**	12.250	**9.447**	**0.121**	**0.379**	**4.285**
MLAN	0.236	7.222	12.862	39.641	18.223	0.155	2.037	11.482
BIGMC	0.208	5.376	6.086	**11.421**	15.502	0.288	2.350	5.890

4) BIGMC_K: The learned partition matrix F_1 for data samples is as the input of the additional clustering method, i.e., K-means, to generate the final clusters. In this case, the reason why we use F_1 not F is that F as a block matrix includes F_1 (the partition matrix of data samples) and F_2 (the partition matrix of anchors).

Fig. 5 shows the clustering performance of BIGMC and its 4 variants on 7 real-world datasets. Fig. 5 (a), (b), and (c) present the performance in terms of ACC, NMI, and F_M, respectively. It can be noted that BIGMC has a better performance than the four variants. This indicates that each component of BIGMC is essential and they can help each other to improve the performance. Specifically, comparing the performance of BIGMC_A and BIGMC_AS, BIGMC_A outperforms BIGMC_AS since the earned unified graph matrix U goes back to improve the initialized similarity matrix S. This also shows the effectiveness of the joint learning strategy.

6.2.4 Convergence Study

To show the effectiveness of the used optimization strategy for the objective function of BIGMC method, we plot the convergence curves of BIGMC over different datasets in Fig. 6. For each sub-figure, the x-axis denotes the number of iterations and the y-axis denotes the objective function value. It can be noticed that BIGMC converges quickly for all datasets. To be more specific, it converges within 5 iterations on 100leaves, Caltech-20, WebKB, and YaleB datasets. It converges within 10 iterations on the other datasets. This indicates that we presented an efficient optimized solution.

6.2.5 Running Time Comparison

The effectiveness of our proposed BIGMC method has been evaluated by all the above experiments. In this section, we aim to explore the efficiency of BIGMC and compare it to that of the state-of-the-art methods. To exclude the influence of initialization, all the algorithms are conducted 5 times and the mean values are shown in Table 4. It can be observed that MVSC method performs the best and BIGMC performs the second on average. Moreover, MVSC, BIGMC, MLAN, and GMC have comparable performance.

6.3 Experiment Summary

In this paper, we assumed that the consensus information contained in multiple views can be represented by a small number of uniform anchor points. Based on this assumption, we proposed BIGMC method. To test the assumption, we examined the performance of BIGMC on both synthetic and real-world data sets. From the experimental results on synthetic datasets in Section 6.1, each learned uniform anchor point (pink square) is the centroid of the corresponding sub-cluster with data points (dots). The learned graphs in the two views (S^1 and S^2), which are constructed by the connections between the uniform anchor points and the data points, were well integrated into a unified graph (U). The U separates the clusters very well since it can learn the consensus information from the two views through the uniform anchor points. From the experimental results on real-world datasets in Section 6.2, our BIGMC method improved the multi-view clustering performance compared to the state-of-art baselines in terms of six metrics (shown in Table 3). Moreover, we showed the adaptive ability of learning the uniform anchor points by BIGMC in a given example in Fig. 4. For different classes, the number of anchors can be learned adaptively, and more anchors can be learned for a class with a larger number of data points. To sum up, we have demonstrated the effectiveness of BIGMC, which implies that the construction of the bipartite graph is efficient. Thus, our assumption is reasonable to some extent.

7 Conclusion

In this paper, we proposed a novel bipartite graph based multi-view clustering (BIGMC) approach. BIGMC jointly learns the similarity graph of each view, the unified bipartite graph, and the representative uniform anchor set in a framework. Moreover, BIGMC adaptively determines the importance of each view and directly obtains the final clusters with a low rank constraint, which is imposed on the unified bipartite Laplacian matrix. Finally, the consensus information are uncovered and the clustering structures are learned through an alternating optimization strategy. The experiments on synthetic and real-world datasets are conducted to demonstrate the effectiveness of BIGMC. In addition, obtaining a globally optimum solution of the objective function is considered as an open problem for our future work.

8 Acknowledgment

This work was supported by the National Science Foundation under grant ECCS 1917275.

References

[1] S. Bickel and T. Scheffer, "Multi-view clustering." in *ICDM*, vol. 4, 2004, pp. 19–26.

[2] L. Li, H. He, J. Li, and G. Yang, "Adversarial domain adaptation via category transfer," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[3] L. Li, Z. Wan, and H. He, "Dual alignment for partial domain adaptation," *IEEE Transactions on Cybernetics*, 2020.

[4] Z. Wan and H. He, "Answernet: Learning to answer questions," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 540–549, 2019.

[5] S. Sun, "A survey of multi-view machine learning," *Neural computing and applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013.

[6] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *Twenty-Third International Joint conference on artificial intelligence*, 2013.

[7] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 252–260.

[8] G. Tzortzis and A. Likas, "Kernel-based weighted multi-view clustering," in *2012 IEEE 12th international conference on data mining*. IEEE, 2012, pp. 675–684.

[9] C.-D. Wang, J.-H. Lai, and S. Y. Philip, "Multi-view clustering based on belief propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 1007–1021, 2015.

[10] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 586–594.

[11] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, "One-step multi-view spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 2022–2034, 2018.

[12] H. Tong, J. He, M. Li, C. Zhang, and W.-Y. Ma, "Graph based multi-modality learning," in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 862–871.

[13] H. Wang, Y. Yang, B. Liu, and H. Fujita, "A study of graph-based system for multi-view clustering," *Knowledge-Based Systems*, vol. 163, pp. 1009–1019, 2019.

[14] Z. Kang, H. Pan, S. C. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE transactions on cybernetics*, 2019.

[15] J. Zhao, L. Li, F. Deng, H. He, and J. Chen, "Discriminant geometrical and statistical alignment with density peaks for domain adaptation," *IEEE Transactions on Cybernetics*, 2020.

[16] H. Wang, Y. Yang, and B. Liu, "Gmc: graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[17] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[18] Z. Wan, H. He, and B. Tang, "A generative model for sparse hyperparameter determination," *IEEE Transactions on Big Data*, vol. 4, no. 1, pp. 2–10, 2018.

[19] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 393–400.

[20] X. Zhao, N. Evans, and J.-L. Dugelay, "A subspace co-training framework for multi-view clustering," *Pattern Recognition Letters*, vol. 41, pp. 73–82, 2014.

[21] D. Guo, J. Zhang, X. Liu, Y. Cui, and C. Zhao, "Multiple kernel learning based multi-view spectral clustering," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 3774–3779.

[22] X. Zhang, B. Chen, H. Sun, Z. Liu, Z. Ren, and Y. Li, "Robust low-rank kernel subspace clustering based on the schatten p-norm and correntropy," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[23] X. Fang, Y. Xu, X. Li, Z. Lai, and W. K. Wong, "Robust semi-supervised subspace clustering via non-negative low-rank representation," *IEEE transactions on cybernetics*, vol. 46, no. 8, pp. 1828–1838, 2015.

[24] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, and Y. Fang,

"Low-rank sparse subspace for spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[25] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[26] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence," *IEEE transactions on knowledge and data engineering*, vol. 29, no. 5, pp. 1129–1143, 2017.

[27] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[28] D. Xie, X. Zhang, Q. Gao, J. Han, S. Xiao, and X. Gao, "Multiview clustering by joint latent representation and similarity learning," *IEEE transactions on cybernetics*, 2019.

[29] T. Zhou, C. Zhang, X. Peng, H. Bhaskar, and J. Yang, "Dual shared-specific multiview subspace clustering," *IEEE transactions on cybernetics*, 2019.

[30] M. Yin, J. Gao, S. Xie, and Y. Guo, "Multiview subspace clustering via tensorial t-product representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 851–864, 2018.

[31] H. Wang, Y. Yang, and T. Li, "Multi-view clustering via concept factorization with local manifold regularization," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 1245–1250.

[32] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognition*, vol. 73, pp. 247–258, 2018.

[33] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1501–1511, 2017.

[34] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 10, pp. 4833–4843, 2018.

[35] R. Wang, F. Nie, Z. Wang, H. Hu, and X. Li, "Parameter-free weighted multi-view projected clustering with structured graph learning," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[36] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[37] 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, 2017.

[38] T. He, Y. Liu, T. H. Ko, K. C. Chan, and Y.-S. Ong, "Contextual correlation preserving multiview featured graph clustering," *IEEE transactions on cybernetics*, 2019.

[39] S. Huang, Z. Xu, I. W. Tsang, and Z. Kang, "Auto-weighted multi-view co-clustering with bipartite graphs," *Information Sciences*, 2019.

[40] F. Nie, C.-L. Wang, and X. Li, "K-multiple-means: A multiple-means clustering method with specified k clusters," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 959–967.

[41] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[42] K. Fan, "On a theorem of weyl concerning eigenvalues of linear transformations i," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 35, no. 11, p. 652, 1949.

[43] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[44] S. Huang, Z. Xu, I. W. Tsang, and Z. Kang, "Auto-weighted multi-view co-clustering with bipartite graphs," *Information Sciences*, vol. 512, pp. 18–30, 2020.

[45] W. Zhuge, F. Nie, C. Hou, and D. Yi, "Unsupervised single and multiple views feature extraction with structured graph," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2347–2359, 2017.

[46] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert systems with applications*, vol. 40, no. 1, pp. 200–210, 2013.

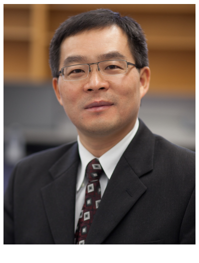
[47] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[48] T. Su and J. G. Dy, "In search of deterministic methods for initializing k-means and gaussian mixture clustering," *Intelligent Data Analysis*, vol. 11, no. 4, pp. 319–338, 2007.

[49] S. Li, L. Li, J. Yan, and H. He, "Sde: A novel clustering framework based on sparsity-density entropy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1575–1587, 2018.

[50] L. Li, H. He, and J. Li, "Entropy-based sampling approaches for multi-class imbalanced problems," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

Lusi Li received B.S. and M.S. degree in computer science from Zhongnan University of Economics and Law, China in 2014 and 2017, respectively. Now, she is currently pursuing a Ph.D degree in electrical engineering from University of Rhode Island (Kingston, RI USA). Her research interests include machine learning, data mining, transfer learning, and multi-view learning.



Haibo He (SM'11-F'18) received the B.S. and M.S.degrees in electrical engineering from the Huazhong University of Science and Technology, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University in 2006. He is currently the Robert Haas Endowed Chair Professor at the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island. His current research interests include computational intelligence, machine learning, data mining, and various applications. He received the IEEE International Conference on Communications Best Paper Award (2014), IEEE CIS Outstanding Early Career Award (2014), and National Science Foundation CAREER Award (2011). He is currently the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems.