This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Efficient Multi-View Clustering via Unified and Discrete Bipartite Graph Learning

Si-Guo Fang, Dong Huang, *Member, IEEE*, Xiao-Sha Cai, Chang-Dong Wang,
Chaobo He, and Yong Tang

***Abstract*— Although previous graph-based multi-view clustering (MVC) algorithms have gained significant progress, most of them are still faced with three limitations. First, they often suffer from high computational complexity, which restricts their applications in large-scale scenarios. Second, they usually perform graph learning either at the single-view level or at the view–consensus level, but often neglect the possibility of the joint learning of single-view and consensus graphs. Third, many of them rely on the *k*-means for discretization of the spectral embeddings, which lack the ability to directly learn the graph with discrete cluster structure. In light of this, this article presents an efficient MVC approach via *u*nified and *d*iscrete *b*ipartite *g*raph *l*earning (UDBGL). Specifically, the anchor-based subspace learning is incorporated to learn the view-specific bipartite graphs from multiple views, upon which the bipartite graph fusion is leveraged to learn a view–consensus bipartite graph with adaptive weight learning. Furthermore, the Laplacian rank constraint is imposed to ensure that the fused bipartite graph has discrete cluster structures (with a specific number of connected components). By simultaneously formulating the view-specific bipartite graph learning, the view–consensus bipartite graph learning, and the discrete cluster structure learning into a unified objective function, an efficient minimization algorithm is then designed to tackle this optimization problem and directly achieve a discrete clustering solution without requiring additional partitioning, which notably has linear time complexity in data size. Experiments on a variety of multi-view datasets demonstrate the robustness and efficiency of our UDBGL approach. The code is available at https://github.com/huangdonghere/UDBGL.**

***Index Terms*— Bipartite graph learning, data clustering, large-scale clustering, linear time, multi-view clustering (MVC).**

Nomenclature

Notations	Descriptions
n	The number of samples
c	The number of clusters
V	The number of views
m	The number of anchors
$d^{(v)}$	The number of features in the vth view
$\mathbf{X}^{(v)} \in \mathbb{R}^{d^{(v)} \times n}$	Data matrix in the vth view
$\mathbf{Z}^{(v)} \in \mathbb{R}^{n \times m}$	View-specific bipartite graph of the vth view
$\mathbf{P} \in \mathbb{R}^{n \times m}$	View–consensus bipartite graph
$\mathbf{A}^{(v)} \in \mathbb{R}^{d^{(v)} \times m}$	Anchor matrix in the vth view
$\boldsymbol{\delta} \in \mathbb{R}^{V}$	The view coefficient vector $\boldsymbol{\delta} = [\delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(V)}]^{\top}$

I. Introduction

With the continuous development of information technology, an increasing amount of data can be collected from multiple sources (or views) in real-world scenarios, which is often referred to as multi-view data [1]. For example, an image can be depicted by different feature descriptors, such as local binary pattern (LBP), histogram of oriented gradient (HOG), and Gabor descriptor. A piece of news can be described in different languages, such Chinese, English, and Spanish. To effectively use the multi-view information for the clustering task, many multi-view clustering (MVC) methods have been developed in the literature, among which the graph-based methods have attracted significant research attention in recent years.

The graph-based MVC methods [2], [3], [4], [5], [6], [7], [8] typically build (or learn) some graph structure(s) to reflect the samplewise relationships in multi-view data, and then partition the graph to obtain the clustering result. A general strategy is to formulate the samplewise relationships into an $n \times n$ adjacent graph, where n is the number of samples. Wang et al. [4] used a mutual reinforcement technique to learn the single-view graphs and a unified global graph. Liang et al. [5], [9] leveraged both the consistency and inconsistency of multiple single-view graphs and learned a unified graph via multi-view graph learning. Besides these graph-learning-based methods [4], [5], [6], [9], another popular subcategory of the graph-based MVC methods are the subspace-learning-based methods [10], [11], [12], [13], [14], [15], which seek to learn a self-representation matrix from low-dimensional subspaces and further use the learned self-representation matrix as a

Manuscript received 9 September 2022; revised 31 January 2023; accepted 21 March 2023. This work was supported in part by the NSFC under Grant 61976097, Grant 62276277, Grant 62077045, and Grant U1811263; in part by the Natural Science Foundation of Guangdong Province under Grant 2021A1515012203; and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022B1515120059. *(Corresponding author: Dong Huang.)*

Si-Guo Fang and Dong Huang are with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China, and also with the Key Laboratory of Smart Agricultural Technology in Tropical South China, Ministry of Agriculture and Rural Affairs, China (e-mail: siguofang@hotmail.com; huangdonghere@gmail.com).

Xiao-Sha Cai and Chang-Dong Wang are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China, and also with the Guangdong Key Laboratory of Information Security Technology, Guangzhou 510006, China (e-mail: xiaoshacai@hotmail.com; changdongwang@hotmail.com).

Chaobo He and Yong Tang are with the School of Computer Science, South China Normal University, Guangzhou 510898, China (e-mail: hechaobo@foxmail.com; ytang@m.scnu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3261460.

Digital Object Identifier 10.1109/TNNLS.2023.3261460

2162-237X © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

global similarity graph for final clustering. Cao et al. [16] proposed a multi-view subspace clustering (SSC) method via smoothness and diversity, with the complementarity of multiple representations investigated. Chen et al. [17] enabled the simultaneous learning of the latent embedding space, the global self-representation, and the cluster structures for multi-view subspace learning. Despite the rapid progress, yet these methods mostly suffer from high computational complexity, where the construction of an $n \times n$ adjacent graph typically consumes $\mathcal{O}(n^2)$ time and the spectral partitioning of this graph may even take $\mathcal{O}(n^3)$ time, which restrict their computational feasibility for large datasets.

To breakthrough the computational bottleneck, the bipartite graph (i.e., anchor graph)- based MVC methods [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28] have recently shown promising capability. Instead of relying on some $n \times n$ graph, the bipartite-graph-based methods typically generate a small set of anchors from the original data and then construct an $n \times m$ bipartite graph to represent the data structure [29], where m is the number of anchors. Specifically, Kang et al. [27] learned a bipartite graph for each view via the anchor-based subspace learning, and then heuristically concatenated the multiple bipartite graphs into a unified one. Wang et al. [26] learned a unified bipartite graph for all the views by means of a set of latent consensus anchors and multiple projection matrices. However, these bipartite graph-based MVC methods [25], [26], [27] either: 1) learn a single bipartite graph for all the views [25], [26] or 2) learn a bipartite graph for each view and then concatenate these multiple bipartite graphs heuristically [27], which lack the ability to simultaneously enforce the view-specific bipartite graph learning and the view–consensus bipartite graph learning. Furthermore, they mostly regard the bipartite graph learning and its spectral partitioning as two separate phases, yet cannot directly obtain the clustering result by learning a unified bipartite graph with discrete cluster structure. More recently, Li et al. [22] proposed a scalable MVC method via bipartite graph fusion. Although it is able to adaptively learn a unified graph with discrete cluster structure by fusing multiple view-specific bipartite graphs, yet these view-specific bipartite graphs are preconstructed and cannot be adaptively optimized. Despite these recent advances, it is still a challenging problem how to enable the unified learning of the view-specific bipartite graphs and the view–consensus bipartite graph, while ensuring the discrete cluster structure of the learned graph and maintaining the efficiency of the overall framework for large-scale datasets.

To address the above-mentioned problem, this article presents an efficient MVC approach via *u*nified and *d*iscrete *b*ipartite *g*raph *l*earning (UDBGL). Particularly, the anchor-based subspace learning is used to learn the view-specific bipartite graphs (for multiple views), which are fused into a view–consensus bipartite graph that in turn promotes the view-specific bipartite graph learning. During the fusion, the view weights are adaptively learned to balance the influences of different view-specific bipartite graphs. Furthermore, we impose a Laplacian rank constraint on the unified bipartite graph so as to directly obtain the desired number of connected components, which leads to the discrete cluster structure without requiring additional partitioning. Notably, the view-specific bipartite graph learning, the view–consensus bipartite graph learning, and the discrete cluster structure learning are seamlessly incorporated into a unified objective function, upon which an efficient alternating optimization algorithm is designed to achieve the final clustering with almost linear computational complexity. Experiments are conducted on ten real-world multi-view datasets, which confirm the superiority of our UDBGL approach over the state-of-the-art.

For clarity, the main contributions of this work are summarized as follows.

1) This article bridges the gap between the view-specific bipartite graph learning and the view–consensus bipartite graph learning, which can be promoted by each other mutually and adaptively during their learning.
2) A unified objective function is presented to encompass the view-specific and view–consensus bipartite graph learning and the discrete cluster structure learning, to solve which *linear-time* optimization algorithm is further designed.
3) An efficient MVC approach termed UDBGL is proposed. Experiments on multiple multi-view datasets demonstrate the clustering robustness and scalability of our UDBGL approach over the state-of-the-art MVC approaches.

The remainder of the article is arranged as follows. Section II reviews the related works. Section III describes the formulation of our UDBGL approach. Section IV presents the optimization algorithm and its theoretical analysis. Section V reports the experimental results. Finally, Section VI concludes this article.

II. Related Work

This article focuses on efficient MVC with unified and discrete bipartite graph learning. Therefore, in this section, our literature review pays attention to three main issues in MVC, namely, how to fuse multiple similarity graphs (from multiple views) into a unified similarity graph, how to build and fuse the similarity graphs with high efficiency (for large datasets), and how to learn the graph with discrete cluster structure without additional partitioning process.

In terms of graph learning (or graph fusion), a variety of MVC methods [3], [4], [5], [7], [9] have been proposed, which aim to fuse the similarity relationships among data samples in multiple views into a unified form. Zhan et al. [3] adopted the rank constraint of the Laplacian matrix and the graph regularization to learn the unified graph from the multiple single-view graphs. Wang et al. [4] fused multiple single-view graphs into a unified one and in turn optimized the single-view graphs through the unified graph. Nie et al. [7] reformulated the standard spectral learning model and proposed a multi-view graph learning framework with adaptive weight learning. Liang et al. [5], [9] simultaneously and explicitly modeled the consistency and inconsistency in multiple single-view graphs and fused them via an efficient optimization algorithm.

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

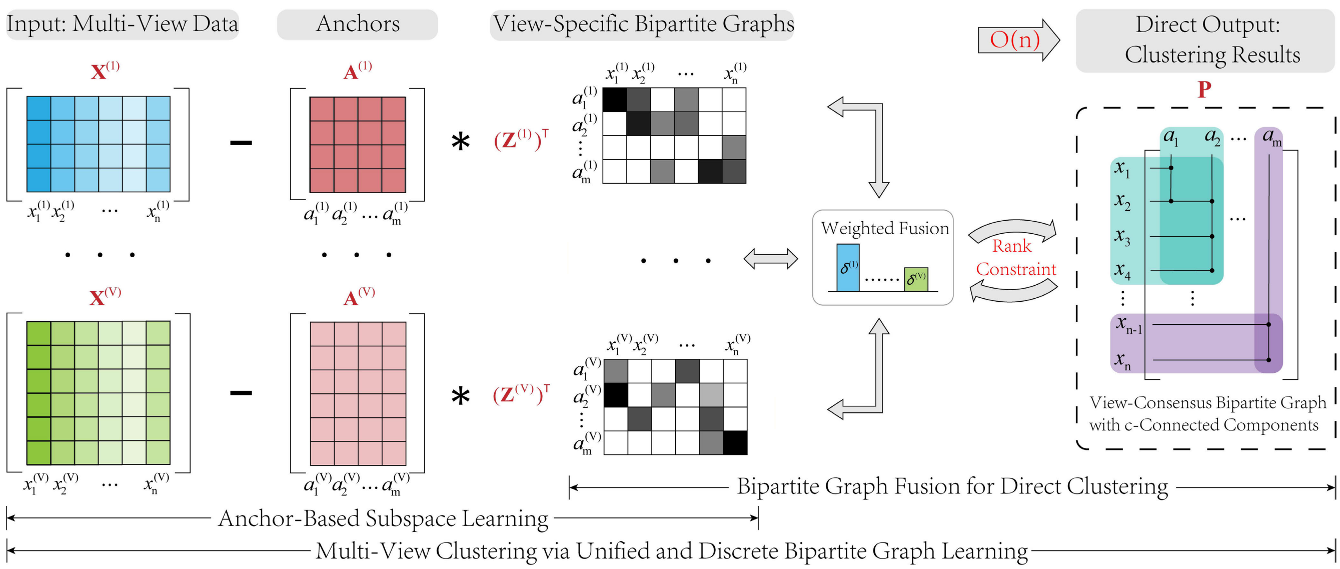

Input: Multi-View Data
Anchors
View-Specific Bipartite Graphs
O(n)
Direct Output: Clustering Results
$X^{(1)}$
$A^{(1)}$
$(Z^{(1)})^T$
$X^{(V)}$
$A^{(V)}$
$(Z^{(V)})^T$
P
Weighted Fusion
$\delta^{(1)}$
$\delta^{(V)}$
Rank Constraint
View-Consensus Bipartite Graph with c-Connected Components
Bipartite Graph Fusion for Direct Clustering
Anchor-Based Subspace Learning
Multi-View Clustering via Unified and Discrete Bipartite Graph Learning

Fig. 1. Illustration of our UDBGL framework. In the anchor-based subspace learning, multiple view-specific bipartite graphs are learned via the self-expressive loss. In the bipartite graph fusion, UDBGL learns the view–consensus bipartite graph with c-connected components and adaptive weights. The view-specific bipartite graph learning (via anchor-based subspace learning) and the view–consensus bipartite graph learning are simultaneously performed and mutually promoted by each other, whose computational complexity is linear to n.

In terms of computational complexity, some bipartite graph-based methods [18], [23], [25], [26], [27], [30] have been proposed, which, instead of building an $n \times n$ graph, resort to an $n \times m$ bipartite graph, where $m \ll n$ is the number of anchors, so as to make them computationally feasible for large-scale datasets. Nie et al. [19] aimed to learn an optimal bipartite graph with ideal connected components for co-clustering. Furthermore, Nie et al. [24] used a dictionary matrix to learn a structured optimal bipartite graph via SSC with the Laplacian rank constraint, which is designed for the single-view situation. For the multi-view situations, Sun et al. [25] proposed a unified bipartite graph learning model with view-unified and orthogonal anchors, where view-specific weights are adaptively learned. In addition, the bipartite graph structure can also be used to reconstruct point-to-point relationships for the incomplete MVC problem [20].

In terms of direct (or discrete) clustering, these aforementioned methods, such as large-scale subspace clustering (LMVSC) [27] and fast parameter-free subspace clustering with consensus anchor guidance (FPMVS-CAG) [26], mostly learn or build a unified graph and then conduct additional spectral partitioning on the graph to obtain the final clustering. However, the spectral partitioning (i.e., spectral clustering) often requires performing the k-means clustering on the spectral embeddings, which may bring in performance degradation due to the instability of k-means. To this end, some previous works incorporate the Laplacian rank constraint to learn a graph with discrete cluster structure, which can directly lead to the clustering result without performing additional partitioning [2], [3], [22], [23], [24], [31], [32]. For example, Nie et al. [2] sought to learn a Laplacian rank constrained graph by considering the similarity matrices of multiple views. Kang et al. [23] performed the multi-view bipartite graph learning with the connectivity constraints to directly learn the clustering structure. But these MVC methods still lack the ability to unify the single-view and the unified bipartite graph learning in a joint learning model. To remedy these issues, in this article, we aim to establish a unified framework with joint view-specific and view–consensus bipartite graph learning and discrete cluster structure learning for large-scale multi-view data.

III. Methodology

In this section, we present the formulation of the proposed UDBGL approach. Specifically, Section III-A introduces the main notations used throughout the article. The view-specific subspace learning with anchors is represented in Section III-B. The Laplacian rank-constrained bipartite graph fusion is formulated in Section III-C. Finally, Section III-D describes the overall framework of UDBGL.

A. Notations

Throughout this article, we denote the scalar values by italic letters, the vectors by boldface lower case letters, and the matrices by boldface capital letters. The jth column of matrix $\mathbf{M}$ is written as $\mathbf{m}_{\cdot j}$ (or $\mathbf{M}(:, j)$), with its ith entry being m_{ij} (or $\mathbf{M}(i, j)$). $\|\mathbf{v}\|_2$ denotes the two-norm of a vector $\mathbf{v}$, and $\|\mathbf{M}\|_F$ denotes the Frobenius norm of matrix $\mathbf{M}$. $\mathbf{v} \geq 0$ (or $\mathbf{M} \geq 0$) represents that all the entries in this vector (or this matrix) are larger than or equal to zero. $\mathbf{I}$ denotes the identity matrix. $\mathbf{1}$ denotes a column vector with all the entries being one. For clarity, The nomenclature table provides the frequently used notations and their descriptions.

B. View-Specific Subspace Learning With Anchors

Let a single-view data matrix be denoted as $\mathbf{X} \in \mathbb{R}^{d\times n}$, which consists of n samples with d features. According to sparse SSC [12], it is assumed that a data point can be written

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

as a linear or affine combination of all other data points, i.e., $\mathbf{X} = \mathbf{X}\mathbf{Z}^\top + \mathbf{E}$, where $\mathbf{Z} \in \mathbb{R}^{n\times n}$ is the self-representation matrix with the constraint $z_{ii} = 0$ to enforce that each point cannot be represented by itself, and $\mathbf{E} \in \mathbb{R}^{d\times n}$ is the error term. It aims to minimize the error term to find a better self-representation matrix, whose objective function can be written as follows:

$$\begin{aligned}&\min_{\mathbf{Z}} \quad \left\|\mathbf{X} - \mathbf{X}\mathbf{Z}^\top\right\|_F^2 + \alpha\|\mathbf{Z}\|_F^2\\ &\text{s.t.} \quad \mathbf{Z} \geq 0, \quad \mathbf{Z}\mathbf{1} = \mathbf{1} \quad \forall i, \quad z_{ii} = 0\end{aligned} \tag{1}$$

where $\|\mathbf{Z}\|_F^2$ is the regularization term that prevents the model from the trivial solution, and $\alpha > 0$ is a hyperparameter to control the influence of the regularization term. By constraining $\mathbf{Z} \geq 0, \mathbf{Z}\mathbf{1} = \mathbf{1}$, the elements in $\mathbf{Z}$ with $0 \leq z_{ij} \leq 1$ can represent the similarity between the ith sample and the jth sample. Thus, $\mathbf{Z}$ can also be viewed as a similarity graph. Note that the elements in the ith row of $\mathbf{Z}$ denote the linear weighting coefficients of the ith sample reconstructed by the other sample points, i.e., $\mathbf{x}_{:i} \approx z_{i1}\mathbf{x}_{:1} + z_{i2}\mathbf{x}_{:2} + \cdots + z_{\text{in}}\mathbf{x}_{:n}$. Here, the row constraint (via $\mathbf{Z}\mathbf{1} = \mathbf{1}$) is used to ensure that each row of $\mathbf{Z}$ has at least one nonzero element.

For multi-view SSC, it is challenging to fuse information from different views due to the potential inconsistency among multiple views [10]. Let $\mathbf{X}^{(v)} \in \mathbb{R}^{d^{(v)}\times n}$ denote the vth data matrix of the multi-view dataset, where $d^{(v)}$ is the number of features in the vth view. Thus, we can perform subspace learning on every single view as follows:

$$\begin{aligned}&\min_{\mathbf{Z}^{(v)}} \quad \sum_{v=1}^{V}\left\{\left\|\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\left(\mathbf{Z}^{(v)}\right)^\top\right\|_F^2 + \alpha\left\|\mathbf{Z}^{(v)}\right\|_F^2\right\}\\ &\qquad\quad + \beta\mathcal{F}\left(\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(V)}\right)\\ &\text{s.t.} \quad \mathbf{Z}^{(v)} \geq 0, \quad \mathbf{Z}^{(v)}\mathbf{1} = \mathbf{1} \quad \forall v, i, \quad z_{ii}^{(v)} = 0\end{aligned} \tag{2}$$

where $\mathcal{F}(\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(V)})$ is a regularization term which enforces the consistency of multiple self-representation matrices $\mathbf{Z}^{(v)} \in \mathbb{R}^{n\times n}$, and $\beta > 0$ is a hyperparameter to control the influence of the regularization term.

The above-mentioned model needs to learn multiple $n \times n$ graphs, which takes at least $\mathcal{O}(n^2)$ time. In the general subspace learning, it is assumed that a sample can be written as a linear or affine combination of all other samples [as shown in objective (2)]. For the self-expressiveness property, there may be much redundant information in the original large-scale dataset. Instead of using the entire dataset as the dictionary, using a smaller number of samples as the dictionary may significantly reduce the computational cost while maintaining sufficient representation ability for the latent subspaces. Here, the small number of representative samples is referred to as the anchors. Recently, it has proved to be an efficient and effective strategy to use the anchor graph (which is a bipartite graph between the original samples and the anchors) rather than the $n \times n$ full graph for large-scale multi-view datasets [22], [27]. Specifically, a straightforward way to construct the anchors is to conduct the k-means clustering on the concatenated features and then split the obtained centers (i.e., anchors) of the concatenated features by views [28]. Let $[\mathbf{A}^{(1)}; \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(V)}] \in \mathbb{R}^{\sum_v d^{(v)}\times m}$ denote the anchor matrix for the dataset $[\mathbf{X}^{(1)}; \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(V)}] \in \mathbb{R}^{\sum_v d^{(v)}\times n}$, where $\mathbf{A}^{(v)}$ is the anchor matrix for the vth view and m is the number of anchors. We extend objective (2) from the conventional subspace learning to the anchor-based subspace learning by replacing the data matrix $\mathbf{X}^{(v)}$ with the anchor matrix $\mathbf{A}^{(v)}$, and thus rewrite the objective function as follows:

$$\begin{aligned}&\min_{\mathbf{Z}^{(v)}} \quad \sum_{v=1}^{V}\left\{\left\|\mathbf{X}^{(v)} - \mathbf{A}^{(v)}\left(\mathbf{Z}^{(v)}\right)^\top\right\|_F^2 + \alpha\left\|\mathbf{Z}^{(v)}\right\|_F^2\right\}\\ &\qquad\quad + \beta\mathcal{F}\left(\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(V)}\right)\\ &\text{s.t.} \quad \mathbf{Z}^{(v)} \geq 0, \quad \mathbf{Z}^{(v)}\mathbf{1} = \mathbf{1}\end{aligned} \tag{3}$$

where the representation matrix $\mathbf{Z}^{(v)} \in \mathbb{R}^{n\times m}$ can be interpreted as a bipartite graph between the n original samples and the m anchors.

C. Rank-Constrained Bipartite Graph Fusion

In this section, we proceed to fuse multiple bipartite graphs from multiple views into a unified bipartite graph. Notably, the multiple bipartite graphs are associated with adaptively learned weights, and a Laplacian rank constraint is further imposed for learning a unified bipartite graph with discrete cluster structure.

Formally, the objective function for bipartite graph fusion can be formulated as follows:

$$\begin{aligned}&\min_{\mathbf{P},\boldsymbol{\delta}} \quad \left\|\sum_{v=1}^{V}\delta^{(v)}\mathbf{Z}^{(v)} - \mathbf{P}\right\|_F^2\\ &\text{s.t.} \quad \boldsymbol{\delta}^\top\mathbf{1} = 1, \quad \boldsymbol{\delta} \geq 0; \mathbf{P} \in \boldsymbol{\Omega}\end{aligned} \tag{4}$$

where $\boldsymbol{\delta} = [\delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(V)}]^\top$ is the view coefficient vector, and $\boldsymbol{\Omega} = \{\mathbf{P} \mid \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P} \geq 0\}$ ensures that $\mathbf{Z}^{(v)}$ and $\mathbf{P}$ are at a similar scale.

After obtaining the unified bipartite graph $\mathbf{P}$, the traditional strategy is to approximately construct a similarity graph among all the samples through the obtained bipartite graph. The doubly stochastic similarity graph $\mathbf{S} \in \mathbb{R}^{n\times n}$ [20], [33], whose summation of each row or each column equals to 1, can be built as follows:

$$\mathbf{S} = \mathbf{P}\boldsymbol{\Lambda}^{-1}\mathbf{P}^\top \tag{5}$$

$$\boldsymbol{\Lambda} = \text{diag}\left(\mathbf{P}^\top\mathbf{1}\right) \in \mathbb{R}^{m\times m} \tag{6}$$

where the diag$(\cdot)$ function returns a diagonal matrix whose main diagonal elements are the corresponding elements of the input vector. Subsequently, spectral clustering can be conducted on the similarity graph $\mathbf{S}$, which takes at least $\mathcal{O}(n^2c)$ time [27], where c is the desired number of clusters. Regarding this, we redefine the similarity graph $\mathbf{S}$ in a different way from the aforementioned (5). That is,

BIGMC的Z

$$\mathbf{S} = \begin{pmatrix} & \mathbf{P}\\ \mathbf{P}^\top & \end{pmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}. \tag{7}$$

Lemma 1: [22], [34] Let $\mathbf{S}$ be an undirected graph with nonnegative elements. The multiplicity of the eigenvalue 0 of the normalized Laplacian matrix $\tilde{\mathbf{L}}_{\mathbf{S}}$ equals the number of connected components in the graph $\mathbf{S}$.

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

To reduce the time complexity and directly achieve the discrete cluster structure without additional partitioning, we impose a rank constraint on the normalized Laplacian matrix (i.e., $\mathrm{rank}(\widetilde{\mathbf{L}}_{\mathbf{S}}) = n + m - c$) of the similarity graph $\mathbf{S}$, where $\widetilde{\mathbf{L}}_{\mathbf{S}} = \mathbf{I} - \mathbf{D}_{\mathbf{S}}^{-(1/2)}\mathbf{S}\mathbf{D}_{\mathbf{S}}^{-(1/2)}$ is the normalized Laplacian matrix, and $\mathbf{D}_{\mathbf{S}} = \mathrm{diag}(\mathbf{S1}) \in \mathbb{R}^{(n+m)\times(n+m)}$ is the degree matrix. According to Lemma 1, the rank constraint enforces that the multiplicity of eigenvalue 0 of $\widetilde{\mathbf{L}}_{\mathbf{S}}$ should be c, which leads to the learned graph $\mathbf{S}$ with exact c connected components.

Furthermore, since $\mathbf{S}$ and $\mathbf{P}$ represent the same graph, the c-connected components of the bipartite graph $\mathbf{P}$ are guaranteed through the c-connected graph $\mathbf{S}$. Thus, the constraints on $\mathbf{P}$ can be rewritten as follows:

$$\mathbf{\Omega} = \{\mathbf{P} \mid \mathbf{P1} = \mathbf{1}, \mathbf{P} \geq 0; \mathrm{rank}\big(\widetilde{\mathbf{L}}_{\mathbf{S}}\big) = n + m - c\}. \quad (8)$$

Through the Laplacian rank constraint, the final clustering result can be directly gained from the bipartite graph $\mathbf{P}$ without additional partitioning. In particular, the bipartite graph $\mathbf{P}$ with the Laplacian low-rank constraint will have a specific number of connected components [22], [34]. By treating the samples in the same connected component as a cluster, the clustering result can then be obtained. Thereby, we rewrite the objective function of the bipartite graph fusion as follows:

$$\begin{aligned} \min_{\mathbf{P},\boldsymbol{\delta}} \quad & \left\| \sum_{v=1}^{V} \delta^{(v)}\mathbf{Z}^{(v)} - \mathbf{P} \right\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\delta}^\top\mathbf{1} = 1, \boldsymbol{\delta} \geq 0; \mathbf{P1} = \mathbf{1}, \mathbf{P} \geq 0; \mathrm{rank}\big(\widetilde{\mathbf{L}}_{\mathbf{S}}\big) = n + m - c. \end{aligned} \quad (9)$$

Both the general graph and the bipartite graph can use the rank constraint to learn the discrete cluster structure, but they are quite different in connectivity and optimization. In terms of connectivity, the general graph constructs different connected components through the similarity between data samples, while in a bipartite graph it should be decided through the anchors whether two data samples in the bipartite graph are connected. That is, the connectivity between two samples should be achieved through one or more anchors. In terms of optimization, the discrete cluster structure learning of a general graph is usually more time-consuming, while using the bipartite graph structure can significantly reduce the computational complexity.

D. MVC via Unified and Discrete Bipartite Graph Learning

Though previous MVC methods based on bipartite graph learning have achieved considerable progress, yet most of them *either* preconstruct the view-specific bipartite graphs and then learn a unified bipartite graph [18], [19], [22], [28] *or* learn multiple view-specific bipartite graphs and then simply concatenate the learned view-specific bipartite graphs into a unified one [27]. However, the preconstructed bipartite graphs lack the ability of adaptive learning and refining, while the simple concatenation cannot well explore the latent structures of multiple bipartite graphs. Surprisingly, there remains a challenging gap between the view-specific bipartite graph learning and the view–consensus bipartite graph learning in most of the previous works [18], [19], [22], [27], [28].

In this article, we seek to unify the view-specific bipartite graph learning, the view–consensus bipartite graph learning, and the discrete cluster structure learning in a joint learning model. Particularly, objective (3) (for anchor-based subspace learning) and objective (9) are jointly formulated, where the graph fusion objective serves as a regularization term for objective (3) and the Laplacian rank constraint bridges the gap between the bipartite graph learning and the discrete cluster structure learning. Formally, the objective function for the unified and discrete bipartite graph learning can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{Z}^{(v)},\mathbf{P},\boldsymbol{\delta}} \quad & \sum_{v=1}^{V} \left\{ \left\| \mathbf{X}^{(v)} - \mathbf{A}^{(v)}\big(\mathbf{Z}^{(v)}\big)^\top \right\|_F^2 + \alpha \left\| \mathbf{Z}^{(v)} \right\|_F^2 \right\} \\ & + \beta \left\| \sum_{v=1}^{V} \delta^{(v)}\mathbf{Z}^{(v)} - \mathbf{P} \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{Z}^{(v)} \geq 0, \mathbf{Z}^{(v)}\mathbf{1} = \mathbf{1}; \boldsymbol{\delta}^\top\mathbf{1} = 1, \boldsymbol{\delta} \geq 0; \mathbf{P1} = \mathbf{1}, \mathbf{P} \geq 0 \\ & \mathrm{rank}(\widetilde{\mathbf{L}}_{\mathbf{S}}) = n + m - c. \end{aligned} \quad (10)$$

In Section IV, we will present a fast optimization algorithm to minimize this unified objective function, which enables the mutual promotion of the view-specific bipartite graphs and the view–consensus bipartite graph and directly achieves the final clustering result without requiring additional graph partitioning.

IV. Optimization and Theoretical Analysis

In this section, we design an efficient alternating optimization algorithm to minimize the objective function (10). Specifically, in each iteration, we update each of the variables $\mathbf{P}$, $\mathbf{Z}^{(v)}$, and $\boldsymbol{\delta}$ while fixing the other variables in Sections IV-A–IV-C, respectively. Furthermore, we provide the computational complexity analysis and the theoretical convergence analysis in Sections IV-D and IV-E, respectively.

A. Update *P*

With the other variables fixed, the subproblem that only relates to $\mathbf{P}$ can be written as follows:

$$\begin{aligned} \min_{\mathbf{P}} \quad & \left\| \sum_{v=1}^{V} \delta^{(v)}\mathbf{Z}^{(v)} - \mathbf{P} \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{P1} = \mathbf{1}, \mathbf{P} \geq 0; \mathrm{rank}\big(\widetilde{\mathbf{L}}_{\mathbf{S}}\big) = n + m - c. \end{aligned} \quad (11)$$

According to [22] and Ky Fan's theorems [35], let $\mathbf{B} = \sum_{v=1}^{V} \delta^{(v)}\mathbf{Z}^{(v)}$, the subproblem (11) can be further transformed into

$$\begin{aligned} \min_{\mathbf{P},\mathbf{F}} \quad & \|\mathbf{B} - \mathbf{P}\|_F^2 + \gamma \ Tr\big(\mathbf{F}^\top\widetilde{\mathbf{L}}_{\mathbf{S}}\mathbf{F}\big) \\ \text{s.t.} \quad & \mathbf{P1} = \mathbf{1}, \quad \mathbf{P} \geq 0; \mathbf{F}^\top\mathbf{F} = \mathbf{I} \end{aligned} \quad (12)$$

where $\mathbf{F} \in \mathbb{R}^{(n+m)\times c}$ is the indicator matrix, and γ can be automatically determined based on the number of connected components of $\mathbf{P}$. Furthermore, we alternately optimize the variables $\mathbf{F}$ and $\mathbf{P}$.

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

First, with $\mathbf{P}$ fixed, the subproblem with respect to $\mathbf{F}$ can be formulated as follows:

$$\begin{aligned}&\min_{\mathbf{F}^\top\mathbf{F}=\mathbf{I}} \quad \mathrm{Tr}\left(\mathbf{F}^\top\widetilde{\mathbf{L}}_{\mathbf{S}}\mathbf{F}\right)\\ \Leftrightarrow &\max_{\mathbf{F}_{(n)}^\top\mathbf{F}_{(n)}+\mathbf{F}_{(m)}^\top\mathbf{F}_{(m)}=\mathbf{I}} Tr\left(\mathbf{F}_{(n)}^\top\mathbf{D}_{(n)}^{-\frac{1}{2}}\mathbf{P}\mathbf{D}_{(m)}^{-\frac{1}{2}}\mathbf{F}_{(m)}\right)\end{aligned} \tag{13}$$

with

$$\mathbf{F}=\begin{pmatrix}\mathbf{F}_{(n)}\\ \mathbf{F}_{(m)}\end{pmatrix},\ \mathbf{F}_{(n)}\in\mathbb{R}^{n\times c},\quad \mathbf{F}_{(m)}\in\mathbb{R}^{m\times c} \tag{14}$$

$$\mathbf{D_S}=\begin{pmatrix}\mathbf{D}_{(n)} & \\ & \mathbf{D}_{(m)}\end{pmatrix},\quad \mathbf{D}_{(n)}\in\mathbb{R}^{n\times n},\quad \mathbf{D}_{(m)}\in\mathbb{R}^{m\times m}. \tag{15}$$

According to [19], the optimal solutions to the problem (13) can be obtained as $\mathbf{F}_{(n)} = ((2)^{1/2}/2)\mathbf{U}$ and $\mathbf{F}_{(m)} = ((2)^{1/2}/2)\mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ are the largest c left and right singular vectors of $\mathbf{D}_{(n)}^{-(1/2)}\mathbf{P}\mathbf{D}_{(m)}^{-(1/2)}$, respectively.

Second, according to [22], we can have the following equality:

$$Tr(\mathbf{F}^\top\widetilde{\mathbf{L}}_{\mathbf{S}}\mathbf{F})=\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\frac{\mathbf{F}_{(n)}(i,:)}{\sqrt{\mathbf{D}_{(n)}(i,i)}}-\frac{\mathbf{F}_{(m)}(j,:)}{\sqrt{\mathbf{D}_{(m)}(j,j)}}\right\|_2^2 p_{ij}. \tag{16}$$

Let $q_{ij} = \|(\mathbf{F}_{(n)}(i,:)/(\mathbf{D}_{(n)}(i,i))^{1/2}) - (\mathbf{F}_{(m)}(j,:)/(\mathbf{D}_{(m)}(j,j))^{1/2})\|_2^2$. With $\mathbf{F}$ fixed, the subproblem with respect to $\mathbf{P}$ can be written as follows:

$$\min_{\mathbf{P1}=\mathbf{1},\mathbf{P}\geq 0}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(b_{ij}-p_{ij}\right)^2+\gamma q_{ij}p_{ij}. \tag{17}$$

Because the optimization of each row of $\mathbf{P}$ in the subproblem (17) is independent, we optimize $\mathbf{P}$ by rows

$$\min_{\mathbf{p}_{i:}\mathbf{1}=1,\mathbf{p}_{i:}\geq 0}\left\|\mathbf{p}_{i:}-\left(\mathbf{b}_{i:}-\frac{\gamma}{2}\mathbf{q}_{i:}\right)\right\|_2^2. \tag{18}$$

Then the closed-form solution $\mathbf{p}_{i:}$ can be obtained according to [36].

B. Update $\mathbf{Z}^{(v)}$

With the other variables fixed, the subproblem that only relates to $\mathbf{Z}^{(v)}$ can be written as follows:

$$\begin{aligned}\min_{\mathbf{Z}^{(v)}} \quad & \left\|\mathbf{X}^{(v)}-\mathbf{A}^{(v)}\left(\mathbf{Z}^{(v)}\right)^\top\right\|_F^2+\alpha\left\|\mathbf{Z}^{(v)}\right\|_F^2\\ & +\beta\left\|\sum_{i=1}^{V}\delta^{(i)}\mathbf{Z}^{(i)}-\mathbf{P}\right\|_F^2\\ \text{s.t.} \quad & \mathbf{Z}^{(v)}\geq 0,\quad \mathbf{Z}^{(v)}\mathbf{1}=\mathbf{1}.\end{aligned} \tag{19}$$

Since the optimization of each row of $\mathbf{Z}^{(v)}$ in the subproblem (19) is independent, we optimize $\mathbf{Z}^{(v)}$ by rows

$$\begin{aligned}\min_{\mathbf{z}_{j:}^{(v)}} \quad & \left\|\mathbf{x}_{:j}^{(v)}-\mathbf{A}^{(v)}\left(\mathbf{z}_{j:}^{(v)}\right)^\top\right\|_2^2+\alpha\left\|\mathbf{z}_{j:}^{(v)}\right\|_2^2\\ & +\beta\left\|\sum_{i=1}^{V}\delta^{(i)}\mathbf{z}_{j:}^{(i)}-\mathbf{p}_{j:}\right\|_2^2\\ \text{s.t.} \quad & \mathbf{z}_{j:}^{(v)}\geq 0,\quad \mathbf{z}_{j:}^{(v)}\mathbf{1}=1.\end{aligned} \tag{20}$$

The above optimization problem (20) can be easily formulated as the following quadratic programming (QP) problem:

$$\begin{aligned}\min_{\mathbf{z}_{j:}^{(v)}} \quad & \mathbf{z}_{j:}^{(v)}\bar{\mathbf{H}}\left(\mathbf{z}_{j:}^{(v)}\right)^\top+\mathbf{z}_{j:}^{(v)}\bar{\mathbf{f}}\\ \text{s.t.} \quad & \mathbf{z}_{j:}^{(v)}\geq 0,\quad \mathbf{z}_{j:}^{(v)}\mathbf{1}=1\end{aligned} \tag{21}$$

where

$$\begin{aligned}\bar{\mathbf{H}} &= \left(\mathbf{A}^{(v)}\right)^\top\mathbf{A}^{(v)}+\left(\alpha+\beta\left(\delta^{(v)}\right)^2\right)\mathbf{I}\\ \bar{\mathbf{f}} &= -2\left(\mathbf{A}^{(v)}\right)^\top\mathbf{x}_{:j}^{(v)}+2\beta\delta^{(v)}\left(\sum_{i\neq v}\delta^{(i)}\left(\mathbf{z}_{j:}^{(i)}\right)^\top-\mathbf{p}_{j:}^\top\right).\end{aligned} \tag{22}$$

The problem (21) can be solved using the augmented Lagrangian multiplier (ALM) [37] (see Section IV-C for details).

Algorithm 1 Multi-View Clustering via Unified and Discrete Bipartite Graph Learning (UDBGL)

```
Input: Multi-view dataset $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$, the number of clusters $c$,
    the number of anchors $m \geq c$, the parameters $\alpha > 0$ and $\beta > 0$.
Preparation: Normalize $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$. Conduct $k$-means on the
    concatenated features $[\mathbf{X}^{(1)}; \mathbf{X}^{(2)}; \cdots; \mathbf{X}^{(V)}]$ to obtain the
    anchors $[\mathbf{A}^{(1)}; \mathbf{A}^{(2)}; \cdots; \mathbf{A}^{(V)}]$.
Initialization: $\boldsymbol{\delta} = \mathbf{1}/V$. Initialize $\mathbf{Z}^{(v)}$ by $K$-nearest-
    neighbor ($K$-NN) bipartite graph.
repeat
    Initialize $\gamma = 0.1$, $\mathbf{D}_{(n)} = diag(\mathbf{B1})$, and $\mathbf{D}_{(m)} = diag(\mathbf{B}^\top\mathbf{1})$.
    Initialize $\mathbf{F}$ by solving problem (13) (via replacing $\mathbf{P}$ with $\mathbf{B}$).
    repeat
        Update $q_{ij} = \left\|\frac{\mathbf{F}_{(n)}(i,:)}{\sqrt{\mathbf{D}_{(n)}(i,i)}}-\frac{\mathbf{F}_{(m)}(j,:)}{\sqrt{\mathbf{D}_{(m)}(j,j)}}\right\|_2^2$.
        Update $\mathbf{P}$ by solving problem (18) and update $\mathbf{D_S}$.
        Update $\mathbf{F}$ by solving problem (13).
        $temp =$ the multiplicity of the eigenvalue 0 of $\widetilde{\mathbf{L}}_{\mathbf{S}}$.
        if $temp < c$ then
            $\gamma \leftarrow 2\gamma$.
        else if $temp > c$ then
            $\gamma \leftarrow \gamma/2$.
        end if
    until Obtaining the $c$-connected bipartite graph $\mathbf{P}$
    $\forall v$, update $\mathbf{Z}^{(v)}$ by solving problem (19).
    Calculate $\mathbf{H} = \hat{\mathbf{Z}}^\top\hat{\mathbf{Z}}$, and $\mathbf{f} = 2\hat{\mathbf{Z}}^\top\hat{\mathbf{p}}$.
    Initialize $\boldsymbol{\delta} = \mathbf{1}/V$, $\mu = 2$, and $\boldsymbol{\eta}$.
    repeat
        Update $\boldsymbol{\rho}$ by (29).
        Update $\boldsymbol{\delta}$ by solving problem (30).
        $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} + \mu(\boldsymbol{\delta} - \boldsymbol{\rho})$,$\mu \leftarrow 2\mu$.
    until $\boldsymbol{\delta}$ converges
until Convergence or maximum iteration reached
Output: The final clustering can be directly obtained from $\mathbf{P}$.
```

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
DETAILS OF THE MULTI-VIEW DATASETS IN OUR EXPERIMENTS

Dataset	#Sample	#View	#Class	Dimension
WebKB-Texas	187	2	5	Citation(187), Content(1,703)
MSRCv1	210	4	7	CM(24), GIST(512), LBP(256), GENT(254)
Out-Scene	2,688	4	8	GIST(512), HOG(432), LBP(256), Gabor(48)
Cora	2,708	4	7	View1(2,708), View2(1,433), View3(2,708), View4(2,708)
Citeseer	3,312	2	6	Content(3,703), Citation(3,312)
Notting-Hill	4,660	3	5	Gabor(6,750),LBP(3,304),Intensity(2,000)
VGGFace	34,027	4	50	LBP(944), HOG(576), GIST(512), Gabor(640)
YoutubeFaces	51,615	4	15	LBP(944), HOG(576), GIST(512), Gabor(640)
CIFAR-10	60,000	4	10	LBP(944), HOG(576), GIST(512), Gabor(640)
CIFAR-100	60,000	4	100	LBP(944), HOG(576), GIST(512), Gabor(640)

C. Update $\boldsymbol{\delta}$

With the other variables fixed, the subproblem that only relates to $\boldsymbol{\delta}$ can be written as follows:

$$\begin{aligned}&\min_{\boldsymbol{\delta}} \left\| \sum_{v=1}^{V} \delta^{(v)}\mathbf{Z}^{(v)} - \mathbf{P} \right\|_F^2 \\ &\text{s.t.} \quad \boldsymbol{\delta}^\top \mathbf{1} = 1, \quad \boldsymbol{\delta} \geq 0. \end{aligned} \tag{23}$$

To solve the subproblem (23), we vectorize each matrix $\mathbf{Z}^{(v)}$ into a vector $\hat{\mathbf{z}}^{(v)}$, that is,

$$\hat{\mathbf{z}}^{(v)} = \left[\mathbf{z}_{:1}^{(v)}; \mathbf{z}_{:2}^{(v)}, \ldots, \mathbf{z}_{:m}^{(v)}\right] \in \mathbb{R}^{nm\times 1} \tag{24}$$

where $\mathbf{z}_{:i}^{(v)}$ represents the ith column of $\mathbf{Z}^{(v)}$. Furthermore, we stack the vectors $\hat{\mathbf{z}}^{(v)}$ for V views into a matrix $\hat{\mathbf{Z}} = [\hat{\mathbf{z}}^{(1)}, \hat{\mathbf{z}}^{(2)}, \ldots, \hat{\mathbf{z}}^{(V)}] \in \mathbb{R}^{nm\times V}$. Note that $\mathbf{B} = \sum_{v=1}^{V} \delta^{(v)}\mathbf{Z}^{(v)}$, and $\hat{\mathbf{b}}$ is the vectorization of $\mathbf{B}$ [similar to (24)], so we have $\hat{\mathbf{b}} = \hat{\mathbf{Z}}\boldsymbol{\delta}$. Similarly, $\hat{\mathbf{p}}$ is the vectorization of $\mathbf{P}$, then the problem (23) can be rewritten as follows:

$$\begin{aligned}&\min_{\boldsymbol{\delta}^\top\mathbf{1}=1,\boldsymbol{\delta}\geq 0} \left\|\hat{\mathbf{Z}}\boldsymbol{\delta} - \hat{\mathbf{p}}\right\|_2^2 \\ \Leftrightarrow &\min_{\boldsymbol{\delta}^\top\mathbf{1}=1,\boldsymbol{\delta}\geq 0} \boldsymbol{\delta}^\top\mathbf{H}\boldsymbol{\delta} - \boldsymbol{\delta}^\top\mathbf{f}\end{aligned} \tag{25}$$

where $\mathbf{H} = \hat{\mathbf{Z}}^\top\hat{\mathbf{Z}}$, and $\mathbf{f} = 2\hat{\mathbf{Z}}^\top\hat{\mathbf{p}}$. Objective (25) is a convex quadratic programming with the semi-definite quadratic matrix $\mathbf{H}$. Here, we use the ALM [37] to optimize objective (25), whose solution is obtained by solving its equivalence

$$\min_{\boldsymbol{\delta}^\top\mathbf{1}=1,\boldsymbol{\delta}\geq 0,\boldsymbol{\delta}=\boldsymbol{\rho}} \boldsymbol{\delta}^\top\mathbf{H}\boldsymbol{\rho} - \boldsymbol{\delta}^\top\mathbf{f}. \tag{26}$$

The augmented Lagrangian function of objective (26) can be written as follows:

$$\min_{\boldsymbol{\delta}^\top\mathbf{1}=1,\boldsymbol{\delta}\geq 0,\boldsymbol{\rho}} \boldsymbol{\delta}^\top\mathbf{H}\boldsymbol{\rho} - \boldsymbol{\delta}^\top\mathbf{f} + \frac{\mu}{2}\left\|\boldsymbol{\delta} - \boldsymbol{\rho} + \frac{1}{\mu}\boldsymbol{\eta}\right\|_2^2 \tag{27}$$

where the optimal solutions $\boldsymbol{\delta}$ and $\boldsymbol{\rho}$ can be optimized alternately. The third term of objective (27) is a penalty term which ensures $\boldsymbol{\delta} = \boldsymbol{\rho}$. The penalty factor $\mu > 0$ is gradually increased in each iteration. Meanwhile, $\boldsymbol{\eta} \in \mathbb{R}^V$ can be updated by $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} + \mu(\boldsymbol{\delta} - \boldsymbol{\rho})$.

First, to update $\boldsymbol{\rho}$ with $\boldsymbol{\delta}$ fixed, the Lagrange function with respect to $\boldsymbol{\rho}$ can be written as follows:

$$\mathcal{L}(\boldsymbol{\rho}) = \boldsymbol{\delta}^{*\top}\mathbf{H}\boldsymbol{\rho} + \frac{\mu}{2}\left\|\boldsymbol{\delta}^* - \boldsymbol{\rho} + \frac{1}{\mu}\boldsymbol{\eta}\right\|_2^2 \tag{28}$$

where $\boldsymbol{\delta}$ is fixed with $\boldsymbol{\delta}^*$. By setting the derivative of $\mathcal{L}(\boldsymbol{\rho})$ with respect to $\boldsymbol{\rho}$ to zero, we can obtain the optimal solution as follows:

$$\boldsymbol{\rho}^* = \boldsymbol{\delta}^* + \frac{1}{\mu}\left(\boldsymbol{\eta} - \mathbf{H}^\top\boldsymbol{\delta}^*\right). \tag{29}$$

Second, to update $\boldsymbol{\delta}$ with fixed $\boldsymbol{\rho}^*$, objective (27) can be transformed into solving the following problem:

$$\min_{\boldsymbol{\delta}^\top\mathbf{1}=1,\boldsymbol{\delta}\geq 0} \left\|\boldsymbol{\delta} - \boldsymbol{\rho}^* + \frac{1}{\mu}(\boldsymbol{\eta} + \mathbf{H}\boldsymbol{\rho}^* - \mathbf{f})\right\|_2^2. \tag{30}$$

Thus, the problem (30) can be solved with a closed-form solution according to [36].

For clarity, the overall algorithm of UDBGL is described in Algorithm 1.

D. Computational Complexity Analysis

In this section, we analyze the computational complexity of the proposed algorithm. The computational cost of UDBGL involves the computation of the preparation, the initialization, and the optimization of three variables. First, the normalization and the k-means take $\mathcal{O}(nd)$ and $\mathcal{O}(ndmt_1)$ time, respectively, where $d = \sum_v d^{(v)}$ and t_1 is the number of k-means iterations. Second, the cost of updating $\mathbf{P}$ is $\mathcal{O}(nmct_2 + nm^2t_2 + m^3t_2)$, where t_2 is the iteration number of updating $\mathbf{P}$. Third, the cost of updating $\mathbf{Z}^{(v)}$ (for all v) is $\mathcal{O}(nm^2d)$. Finally, the cost of updating $\boldsymbol{\delta}$ is $\mathcal{O}(V^2nm)$. Since $n \gg m$, and c, V, t_2 are small constants, the time complexity of UDBGL is linear in n.

E. Convergence Analysis

In this section, we analyze the convergence property of the proposed optimization algorithm. In Section IV-A, we can obtain the optimal solutions for the subproblems of $\mathbf{F}$ and $\mathbf{P}$. To obtain a bipartite graph with a specific number of connected components, we dynamically adjust the parameter γ, which leads to a bipartite graph with discrete cluster structure but may undermine the monotonicity of the objective function of the subproblem (11). For updating $\mathbf{Z}^{(v)}$, the QP problem (21) can be proved to be convex. For any nonzero vector $\bar{\mathbf{x}}$, we have

$$\begin{aligned}\bar{\mathbf{x}}^\top\bar{\mathbf{H}}\bar{\mathbf{x}} &= \bar{\mathbf{x}}^\top\left\{\left(\mathbf{A}^{(v)}\right)^\top\mathbf{A}^{(v)} + \left(\alpha + \beta\left(\delta^{(v)}\right)^2\right)\mathbf{I}\right\}\bar{\mathbf{x}} \\ &= \left\|\mathbf{A}^{(v)}\bar{\mathbf{x}}\right\|_2^2 + \left(\alpha + \beta\left(\delta^{(v)}\right)^2\right)||\bar{\mathbf{x}}||_2^2 \geq 0.\end{aligned} \tag{31}$$

Thus, the matrix of quadratic form is positive semi-definite, and the objective function of the QP problem (21) is convex.

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Note that the constraint functions are affine. Therefore, the QP problem (21) is convex. Similarly, updating $\boldsymbol{\delta}$ is also a convex optimization problem. To summarize, except for the dynamic adjustment of the parameter γ in the subproblem (11), the optimal solutions for all the other subproblems can be achieved. In the following, the empirical convergence of UDBGL will also be verified in Section V-E.

V. Experiments

In this section, we evaluate the proposed UDBGL method against the state-of-the-art MVC methods on ten multi-view datasets. All the experiments are conducted on a PC with 16-GB RAM and an Intel i5-6600 CPU.

A. Datasets and Evaluation Metrics

In the experimental evaluation, ten real-world multi-view datasets are used, namely, WebKB-Texas [38], MSRCv1 [17], Out-Scene [39], Cora [40], Citeseer [40], Notting-Hill, VGGFace,[1] YoutubeFaces, CIFAR-10, and CIFAR-100. Note that CIFAR-10 and CIFAR-100 are two versions of the CIFAR[2] dataset. The details of these datasets are given in Table I.

The clustering results are evaluated by three widely used metrics, i.e., the normalized mutual information (NMI) [41], the accuracy (ACC) [42], and the purity (PUR) [27]. For all the three evaluation metrics, larger values indicate better clustering performance.

B. Baseline Methods and Experimental Settings

We experimentally compare the proposed UDBGL method against the 11 baseline MVC methods, namely, multi-view spectral clustering (MVSC) [18] auto-weighted multiple graph learning (AMGL) [7], MVC with adaptive neighbors (MLAN) [31], self-weighted multiview clustering (SwMC) [2], MVC on topological manifold (MVCTM) [32], scalableand parameter-free multi-view graph clustering (SFMC) [22], LMVSC [27], multi-view structured graph learning (MSGL) [23], scalable multi-view subspace clustering (SMVSC) [25], FPMVS-CAG [26], and efficient one-pass multi-view SSC with consensus anchors (EOMSC-CA) [43].

Among these baseline methods, AMGL, MLAN, SwMC, and MVCTM are four (general) graph-learning-based methods, while MVSC, SFMC, LMVSC, MSGL, SMVSC, FPMVS-CAG, and EOMSC-CA are seven bipartite-graph-based methods. For the proposed method and the baseline methods, the experiments on each dataset are repeated 20 times, and their average scores and standard deviations will be reported. The hyperparameters in the proposed method and the baseline methods are tuned in the range of $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, unless the value (or range) of the hyperparameter is specified by the corresponding paper. The number of anchors is tuned in the range of $\{c, 50, 100, 200\}$. To avoid the expensive computational costs of parameter tuning on the whole large-scale datasets, for all the test methods, if $n > 10\,000$, then we will tune the hyperparameters using a random subset of $10\,000$ samples.

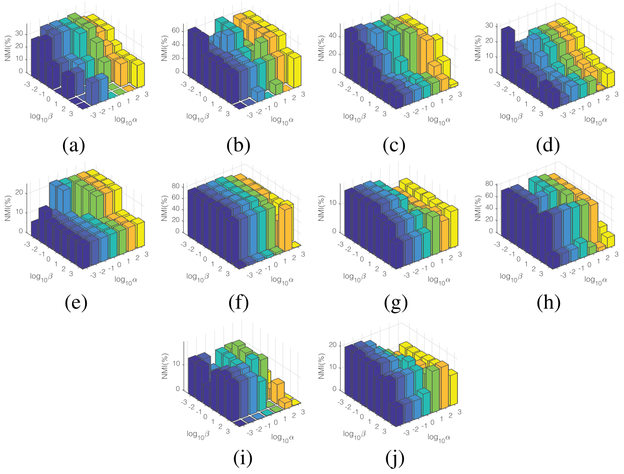


Fig. 2. Clustering performance (with respect to NMI) by UDBGL with varying values of parameters α and β. (a) WebKB-Texas. (b) MSRC-v1. (c) Out-Scene. (d) Cora. (e) Citeseer. (f) Notting-Hill. (g) VGGFace. (h) YoutubeFaces. (i) CIFAR-10. (j) CIFAR-100.

C. Comparison With Other MVC Methods

In this section, we compare the proposed UDBGL method against the 11 baseline methods on the benchmark datasets. Specifically, the clustering scores with respect to NMI, ACC, and PUR by different MVC methods are reported in Tables II–IV, respectively. The best score on each dataset is highlighted in bold, and "N/A" indicates that this method is not computationally feasible on this dataset due to the out-of-memory error. As shown in Table II, in terms of NMI, our UDBGL method outperforms all the baseline methods on nine out of the ten datasets. Particularly, in comparison to the general graph-based methods (i.e., AMGL, MLAN, SwMC, and MVCTM), the bipartite-graph-based methods can generally handle the large datasets more efficiently and produce more robust clustering results in most cases. Among the bipartite graph-based methods (i.e., MVSC, SFMC, LMVSC, MSGL, SMVSC, FPMVS-CAG, EOMSC-CA, and UDBGL), the proposed UDBGL method is able to produce consistently better clustering results on most of the datasets. Although the FPMVS-CAG method yields a higher NMI score than UDBGL on the Out-Scene dataset, yet the UDBGL method outperforms FPMVS-CAG on all the other datasets. Similar advantages of UDBGL with respect to ACC and PUR can also be observed in Tables III and IV, respectively.

Furthermore, we report the average scores and the average ranks (across all the benchmark datasets) by different MVC methods in Tables II–IV. Note that among the 12 MVC methods, if only seven MVC methods are computational feasible and the other five methods are infeasible due to the out-of-memory error on a dataset, then the five infeasible methods will be equally ranked in the eighth position on this dataset. In terms of the average score, UDBGL achieves the average NMI(%), ACC(%), and PUR(%) scores of 43.60, 50.71, and 54.74, respectively, which substantially outperforms the second best average NMI(%), ACC(%), and PUR(%) scores of 36.20, 46.73, and 50.33, respectively. In terms of the

[1] https://www.robots.ox.ac.U.K./~vgg/data/vgg_face2/

[2] https://www.cs.toronto.edu/~kriz/cifar.html

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II

AVERAGE NMI(%) OVER 20 RUNS BY DIFFERENT MVC METHODS. THE BEST RESULT IS HIGHLIGHTED IN **BOLD**

Datasets	MVSC	AMGL	MLAN	SwMC	MVCTM	SFMC	LMVSC	MSGL	SMVSC	FPMVS-CAG	EOMSC-CA	**UDBGL**
WebKB-Texas	1.98$_{\pm 0.25}$	9.89$_{\pm 2.61}$	7.11$_{\pm 1.72}$	9.37$_{\pm 0.00}$	28.74$_{\pm 0.00}$	6.25$_{\pm 0.00}$	20.86$_{\pm 0.00}$	4.56$_{\pm 4.49}$	21.42$_{\pm 0.00}$	22.76$_{\pm 0.00}$	18.84$_{\pm 0.00}$	**37.14**$_{\pm 0.00}$
MSRCv1	49.34$_{\pm 3.78}$	58.84$_{\pm 6.33}$	73.46$_{\pm 0.37}$	62.69$_{\pm 0.00}$	68.91$_{\pm 0.00}$	52.35$_{\pm 0.00}$	24.60$_{\pm 0.00}$	26.98$_{\pm 9.18}$	61.09$_{\pm 0.00}$	56.62$_{\pm 0.00}$	44.65$_{\pm 0.00}$	**77.07**$_{\pm 0.00}$
Out-Scene	23.39$_{\pm 14.73}$	44.19$_{\pm 3.96}$	37.38$_{\pm 0.00}$	43.69$_{\pm 0.00}$	47.29$_{\pm 0.00}$	39.72$_{\pm 0.00}$	45.97$_{\pm 0.00}$	15.10$_{\pm 8.52}$	51.67$_{\pm 0.00}$	**53.04**$_{\pm 0.00}$	38.09$_{\pm 0.00}$	52.45$_{\pm 0.00}$
Cora	0.25$_{\pm 0.09}$	9.00$_{\pm 4.79}$	2.55$_{\pm 0.00}$	8.69$_{\pm 0.01}$	5.47$_{\pm 0.00}$	4.79$_{\pm 0.00}$	12.82$_{\pm 0.00}$	0.54$_{\pm 0.81}$	28.73$_{\pm 0.00}$	18.19$_{\pm 0.00}$	6.56$_{\pm 0.00}$	**30.67**$_{\pm 0.00}$
Citeseer	0.40$_{\pm 0.11}$	0.53$_{\pm 0.06}$	1.85$_{\pm 4.51}$	1.22$_{\pm 0.00}$	1.68$_{\pm 0.00}$	2.43$_{\pm 0.00}$	8.75$_{\pm 0.00}$	0.77$_{\pm 1.20}$	17.86$_{\pm 0.00}$	14.73$_{\pm 0.00}$	14.88$_{\pm 0.00}$	**24.06**$_{\pm 0.00}$
Notting-Hill	53.98$_{\pm 19.66}$	8.19$_{\pm 2.82}$	55.50$_{\pm 0.00}$	8.77$_{\pm 0.00}$	47.95$_{\pm 0.00}$	84.11$_{\pm 0.00}$	68.06$_{\pm 0.00}$	65.20$_{\pm 19.10}$	66.64$_{\pm 0.00}$	67.15$_{\pm 0.00}$	75.13$_{\pm 0.00}$	**87.17**$_{\pm 0.00}$
VGGFace	N/A	N/A	N/A	N/A	N/A	7.79$_{\pm 0.00}$	13.05$_{\pm 0.00}$	2.49$_{\pm 2.30}$	11.16$_{\pm 0.00}$	12.33$_{\pm 0.00}$	12.20$_{\pm 0.00}$	**14.27**$_{\pm 0.00}$
YoutubeFaces	N/A	N/A	N/A	N/A	N/A	78.06$_{\pm 0.00}$	74.91$_{\pm 0.00}$	67.08$_{\pm 2.83}$	71.14$_{\pm 0.00}$	73.54$_{\pm 0.00}$	63.94$_{\pm 0.00}$	**78.69**$_{\pm 0.00}$
CIFAR-10	N/A	N/A	N/A	N/A	N/A	1.76$_{\pm 0.00}$	12.93$_{\pm 0.00}$	0.72$_{\pm 0.46}$	18.37$_{\pm 0.00}$	18.41$_{\pm 0.00}$	16.46$_{\pm 0.00}$	**19.60**$_{\pm 0.00}$
CIFAR-100	N/A	N/A	N/A	N/A	N/A	12.25$_{\pm 0.00}$	**14.89**$_{\pm 0.00}$	1.52$_{\pm 1.14}$	13.93$_{\pm 0.00}$	13.75$_{\pm 0.00}$	13.78$_{\pm 0.00}$	**14.89**$_{\pm 0.00}$
Avg.score	-	-	-	-	-	28.95	29.68	18.50	36.20	35.05	30.45	**43.60**
Avg.rank	9.70	7.90	7.80	7.70	6.70	6.30	4.60	8.90	3.80	3.70	5.70	**1.10**

* Note that N/A indicates the out-of-memory error, and the symbol "-" indicates that the average score cannot be calculated.

TABLE III

AVERAGE ACC(%) OVER 20 RUNS BY DIFFERENT MVC METHODS. THE BEST RESULT IS HIGHLIGHTED IN **BOLD**

Datasets	MVSC	AMGL	MLAN	SwMC	MVCTM	SFMC	LMVSC	MSGL	SMVSC	FPMVS-CAG	EOMSC-CA	**UDBGL**
WebKB-Texas	55.13$_{\pm 0.30}$	51.79$_{\pm 5.36}$	49.63$_{\pm 0.59}$	54.55$_{\pm 0.00}$	63.64$_{\pm 0.00}$	40.11$_{\pm 0.00}$	60.96$_{\pm 0.00}$	45.40$_{\pm 9.58}$	56.68$_{\pm 0.00}$	57.75$_{\pm 0.00}$	57.22$_{\pm 0.00}$	**69.52**$_{\pm 0.00}$
MSRCv1	56.98$_{\pm 5.33}$	64.90$_{\pm 8.10}$	73.81$_{\pm 0.00}$	70.48$_{\pm 0.00}$	**77.62**$_{\pm 0.00}$	56.67$_{\pm 0.00}$	34.29$_{\pm 0.00}$	39.31$_{\pm 9.57}$	67.14$_{\pm 0.00}$	60.00$_{\pm 0.00}$	51.90$_{\pm 0.00}$	74.76$_{\pm 0.00}$
Out-Scene	31.82$_{\pm 10.36}$	49.91$_{\pm 5.12}$	40.55$_{\pm 0.00}$	45.46$_{\pm 0.00}$	50.97$_{\pm 0.00}$	47.84$_{\pm 0.00}$	59.86$_{\pm 0.00}$	26.69$_{\pm 6.22}$	63.32$_{\pm 0.00}$	63.58$_{\pm 0.00}$	52.27$_{\pm 0.00}$	**67.78**$_{\pm 0.00}$
Cora	30.19$_{\pm 0.11}$	33.58$_{\pm 3.00}$	31.39$_{\pm 0.00}$	34.46$_{\pm 0.13}$	31.57$_{\pm 0.00}$	28.32$_{\pm 0.00}$	33.79$_{\pm 0.00}$	27.17$_{\pm 4.29}$	**53.29**$_{\pm 0.00}$	44.53$_{\pm 0.00}$	28.88$_{\pm 0.00}$	42.80$_{\pm 0.00}$
Citeseer	21.24$_{\pm 0.17}$	21.48$_{\pm 0.06}$	22.64$_{\pm 4.29}$	21.62$_{\pm 0.00}$	22.07$_{\pm 0.00}$	24.67$_{\pm 0.00}$	29.35$_{\pm 0.00}$	22.18$_{\pm 1.98}$	41.94$_{\pm 0.00}$	37.53$_{\pm 0.00}$	32.00$_{\pm 0.00}$	**49.03**$_{\pm 0.00}$
Notting-Hill	62.93$_{\pm 14.53}$	36.29$_{\pm 1.84}$	59.21$_{\pm 0.00}$	33.88$_{\pm 0.00}$	41.52$_{\pm 0.00}$	**91.89**$_{\pm 0.00}$	73.76$_{\pm 0.00}$	69.81$_{\pm 14.87}$	71.61$_{\pm 0.00}$	71.31$_{\pm 0.00}$	81.97$_{\pm 0.00}$	80.11$_{\pm 0.00}$
VGGFace	N/A	N/A	N/A	N/A	N/A	8.11$_{\pm 0.00}$	11.02$_{\pm 0.00}$	4.26$_{\pm 1.75}$	9.83$_{\pm 0.00}$	10.74$_{\pm 0.00}$	10.32$_{\pm 0.00}$	**12.31**$_{\pm 0.00}$
YoutubeFaces	N/A	N/A	N/A	N/A	N/A	71.64$_{\pm 0.00}$	64.93$_{\pm 0.00}$	60.87$_{\pm 3.41}$	66.18$_{\pm 0.00}$	62.02$_{\pm 0.00}$	60.53$_{\pm 0.00}$	**71.92**$_{\pm 0.00}$
CIFAR-10	N/A	N/A	N/A	N/A	N/A	11.52$_{\pm 0.00}$	25.01$_{\pm 0.00}$	11.76$_{\pm 0.67}$	29.20$_{\pm 0.00}$	29.55$_{\pm 0.00}$	28.25$_{\pm 0.00}$	**29.84**$_{\pm 0.00}$
CIFAR-100	N/A	N/A	N/A	N/A	N/A	6.87$_{\pm 0.00}$	8.12$_{\pm 0.00}$	1.82$_{\pm 0.48}$	8.14$_{\pm 0.00}$	**9.06**$_{\pm 0.00}$	8.28$_{\pm 0.00}$	**9.06**$_{\pm 0.00}$
Avg.score	-	-	-	-	-	38.76	40.11	30.93	46.73	44.61	41.16	**50.71**
Avg.rank	8.70	8.20	7.90	7.90	6.70	6.80	4.90	8.70	3.70	3.50	5.40	**1.50**

TABLE IV

AVERAGE PUR(%) OVER 20 RUNS BY DIFFERENT MVC METHODS. THE BEST RESULT IS HIGHLIGHTED IN **BOLD**

Datasets	MVSC	AMGL	MLAN	SwMC	MVCTM	SFMC	LMVSC	MSGL	SMVSC	FPMVS-CAG	EOMSC-CA	**UDBGL**
WebKB-Texas	56.10$_{\pm 0.16}$	60.00$_{\pm 1.98}$	56.58$_{\pm 0.88}$	59.89$_{\pm 0.00}$	70.05$_{\pm 0.00}$	55.61$_{\pm 0.00}$	62.57$_{\pm 0.00}$	56.36$_{\pm 2.41}$	67.91$_{\pm 0.00}$	65.78$_{\pm 0.00}$	59.89$_{\pm 0.00}$	**72.73**$_{\pm 0.00}$
MSRCv1	59.31$_{\pm 5.06}$	68.10$_{\pm 7.17}$	**80.48**$_{\pm 0.00}$	73.33$_{\pm 0.00}$	77.62$_{\pm 0.00}$	59.52$_{\pm 0.00}$	40.00$_{\pm 0.00}$	40.95$_{\pm 9.27}$	69.05$_{\pm 0.00}$	62.86$_{\pm 0.00}$	56.67$_{\pm 0.00}$	**80.48**$_{\pm 0.00}$
Out-Scene	32.03$_{\pm 10.40}$	50.47$_{\pm 5.32}$	40.85$_{\pm 0.00}$	45.57$_{\pm 0.00}$	51.04$_{\pm 0.00}$	48.21$_{\pm 0.00}$	59.86$_{\pm 0.00}$	28.06$_{\pm 6.47}$	66.00$_{\pm 0.00}$	67.08$_{\pm 0.00}$	52.27$_{\pm 0.00}$	**67.78**$_{\pm 0.00}$
Cora	30.34$_{\pm 0.08}$	35.54$_{\pm 3.03}$	31.79$_{\pm 0.00}$	36.70$_{\pm 0.09}$	33.64$_{\pm 0.00}$	32.90$_{\pm 0.00}$	38.88$_{\pm 0.00}$	30.43$_{\pm 0.36}$	53.66$_{\pm 0.00}$	45.49$_{\pm 0.00}$	34.34$_{\pm 0.00}$	**53.84**$_{\pm 0.00}$
Citeseer	21.39$_{\pm 0.11}$	21.57$_{\pm 0.05}$	22.80$_{\pm 4.39}$	22.04$_{\pm 0.00}$	22.61$_{\pm 0.00}$	25.30$_{\pm 0.00}$	31.76$_{\pm 0.00}$	22.32$_{\pm 2.09}$	47.01$_{\pm 0.00}$	42.81$_{\pm 0.00}$	45.35$_{\pm 0.00}$	**51.15**$_{\pm 0.00}$
Notting-Hill	65.24$_{\pm 13.91}$	37.07$_{\pm 2.10}$	**92.85**$_{\pm 0.00}$	37.10$_{\pm 0.00}$	71.63$_{\pm 0.00}$	91.89$_{\pm 0.00}$	81.20$_{\pm 0.00}$	75.37$_{\pm 13.29}$	78.41$_{\pm 0.00}$	78.50$_{\pm 0.00}$	84.70$_{\pm 0.00}$	87.23$_{\pm 0.00}$
VGGFace	N/A	N/A	N/A	N/A	N/A	8.52$_{\pm 0.00}$	11.84$_{\pm 0.00}$	4.43$_{\pm 1.85}$	10.28$_{\pm 0.00}$	11.15$_{\pm 0.00}$	11.17$_{\pm 0.00}$	**13.29**$_{\pm 0.00}$
YoutubeFaces	N/A	N/A	N/A	N/A	N/A	78.94$_{\pm 0.00}$	72.02$_{\pm 0.00}$	66.18$_{\pm 2.50}$	69.99$_{\pm 0.00}$	71.90$_{\pm 0.00}$	62.61$_{\pm 0.00}$	**79.48**$_{\pm 0.00}$
CIFAR-10	N/A	N/A	N/A	N/A	N/A	11.92$_{\pm 0.00}$	27.65$_{\pm 0.00}$	11.91$_{\pm 0.75}$	32.40$_{\pm 0.00}$	**32.85**$_{\pm 0.00}$	30.57$_{\pm 0.00}$	31.61$_{\pm 0.00}$
CIFAR-100	N/A	N/A	N/A	N/A	N/A	7.57$_{\pm 0.00}$	9.66$_{\pm 0.00}$	1.89$_{\pm 0.53}$	8.64$_{\pm 0.00}$	9.45$_{\pm 0.00}$	8.86$_{\pm 0.00}$	**9.83**$_{\pm 0.00}$
Avg.score	-	-	-	-	-	42.04	43.54	33.79	50.33	48.79	44.64	**54.74**
Avg.rank	9.70	8.00	7.00	7.80	6.80	6.50	4.70	8.80	3.90	3.80	5.40	**1.40**

average rank, UDBGL obtains the average ranks (with respect to NMI, ACC, and PUR) of 1.10, 1.50, and 1.40, respectively, while the second best method only obtains the average ranks of 3.70, 3.50, and 3.80, respectively, which demonstrate the robustness of the proposed UDBGL method over the other MVC methods.

D. Parameter Analysis

In this section, we test the influence of the hyperparameters α and β as well as the number of anchors m in the proposed UDBGL method.

In UDBGL, the two hyperparameters, i.e., α and β, are used to negotiate the importance of the regularization term (of $\mathbf{Z}^{(v)}$) and the graph fusion term. The clustering performance of UDBGL with varying hyperparameters α and β is illustrated in Fig. 2, which shows that moderate parameter values are generally beneficial to the clustering performance on various datasets.

Then, we test the clustering performance of the proposed UDBGL method with the number of anchors varying in the range of $\{c, 50, 100, 200\}$, which corresponds to the four bars in each subfigure of Fig. 3. Since the anchor number should be greater than or equal to the number of clusters (i.e., c), the number of tested values may be less than four on some datasets, e.g., on the CIFAR-100 dataset (with $c = 100$) only two values (i.e., 100 and 200) are tested, corresponding to the two bars in Fig. 3(j). As can be observed in Fig. 3, setting the anchor number equal to the cluster number c can lead to quite robust clustering results on most of the datasets.

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

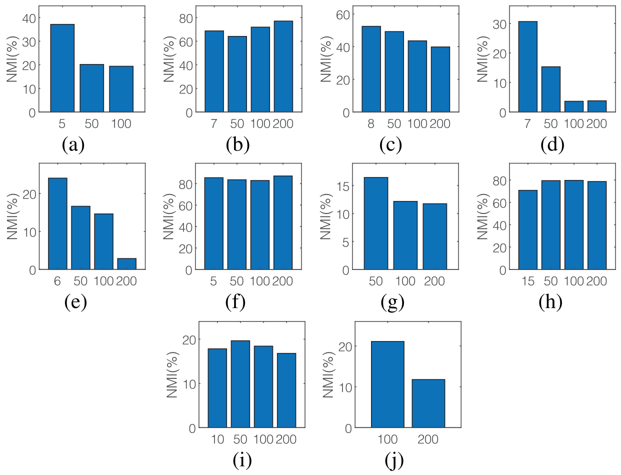


Fig. 3. Clustering performance (with respect to NMI) by UDBGL with varying number of anchors. (a) WebKB-Texas. (b) MSRC-v1. (c) Out-Scene. (d) Cora. (e) Citeseer. (f) Notting-Hill. (g) VGGFace. (h) YoutubeFaces. (i) CIFAR-10. (j) CIFAR-100.

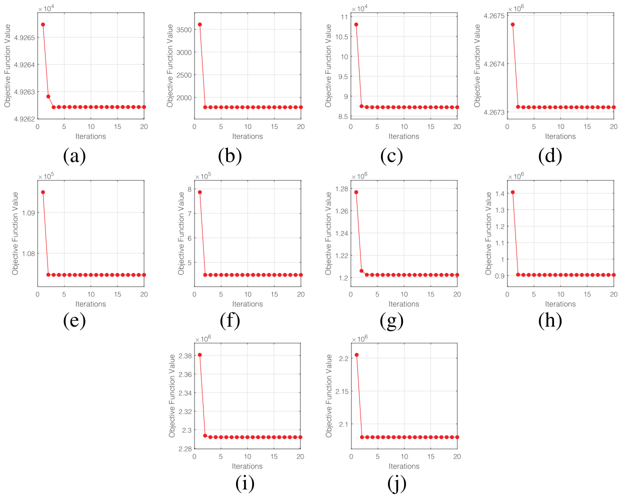


Fig. 4. Convergence of the objective value of UDBGL with increasing iterations. (a) WebKB-Texas. (b) MSRC-v1. (c) Out-Scene. (d) Cora. (e) Citeseer. (f) Notting-Hill. (g) VGGFace. (h) YoutubeFaces. (i) CIFAR-10. (j) CIFAR-100.

E. Empirical Convergence Analysis

In this section, we test the convergence of UDBGL on the benchmark datasets. Fig. 4 shows the convergence curves of the objective function values of UDBGL after different iterations. As can be seen in Fig. 4, the objective function value decreases very fast and generally converges within 20 iterations on various datasets, which shows the good convergence property of the proposed UDBGL method.

F. Ablation Study

In this section, we test the influence of the single-phase formulation of UDBGL by comparing it with the two-phase versions. Specifically, we design two variants by removing the view-specific graph learning part or by splitting the model into two phases in the ablation study. Our first variant is to first use the K-nearest neighbor (K-NN) graphs as the preconstructed graphs and then perform graph fusion to learn a view–consensus graph with discrete cluster structure, which can be viewed as removing the view-specific graph learning part from UDBGL. The second variant is to treat the view-specific graph learning and the view–consensus graph learning in UDBGL as two separate phases, which first enforces subspace learning to obtain the view-specific bipartite graphs and then performs graph fusion to achieve the final clustering. As shown in Fig. 5, the simultaneous learning of the view-specific graphs and the view–consensus graph can generally lead to more robust clustering performance than the two variants across the ten benchmark datasets. In particular, the clustering performances of the two variants have shown instability across different datasets, which means that they perform well on some datasets but may significantly decline on some other datasets (e.g., on the WebKB-Texas, MSRC-v1, Cora, Notting-Hill, and CIFAR-10 datasets). In comparison to the two variants, the unified formulation of UDBGL achieves consistently high-quality clustering performance across the ten datasets, which is probably due to its ability to learn the view-specific graphs and the view–consensus graph jointly and adaptively.

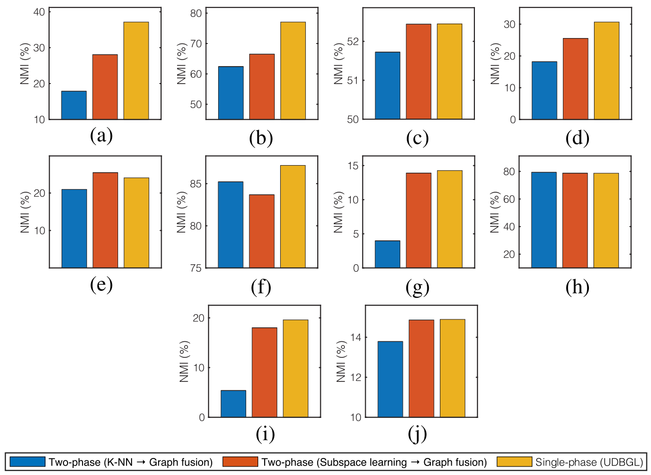


Fig. 5. Ablation analysis on the benchmark datasets. (a) WebKB-Texas. (b) MSRC-v1. (c) Out-Scene. (d) Cora. (e) Citeseer. (f) Notting-Hill. (g) VGGFace. (h) YoutubeFaces. (i) CIFAR-10. (j) CIFAR-100.

G. Execution Time

In this section, we report the execution times of our UDBGL method and the baseline methods on the eight larger datasets (with $n > 2000$). As shown in Fig. 6, MVSC, AMGL, MLAN, SwMC, and MVCTM methods are not computationally feasible on the four large-scale datasets (with over 30 000 samples), while the other seven bipartite-graph-based methods are computationally feasible on all the datasets. Particularly, the proposed UDBGL method has shown comparable time cost to the MSGL, SMVSC, and FPMVS-CAG methods on most of the benchmark datasets. From the performance results in Tables II–IV and the execution times in Fig. 6, we can observe that our UDBGL method is able to achieve advantageous clustering performance on a variety of datasets while maintaining highly competitive efficiency.

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

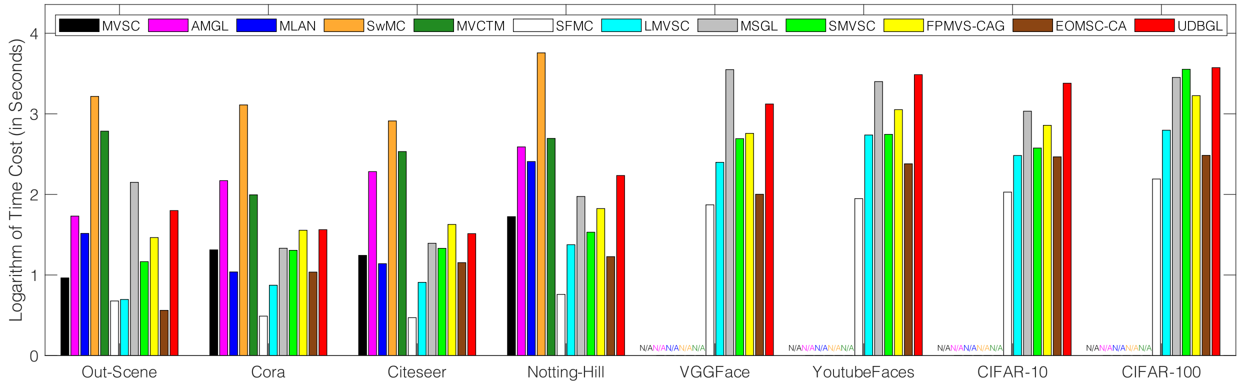


Fig. 6. Execution times of different MVC methods on the eight larger datasets (with $n > 2000$). Note that N/A indicates the out-of-memory error.

VI. Conclusion

In this article, we propose an efficient MVC approach termed UDBGL. In particular, the anchor-based subspace learning is enforced to learn the view-specific bipartite graphs from multiple views, which are fused into a view–consensus bipartite graph with adaptive view weights. Remarkably, the view-specific bipartite graph learning and the view–consensus bipartite graph learning can promote each other mutually and adaptively, which are further coupled with the Laplacian low-rank constraint to directly obtain the discrete cluster structure from the fused graph. An alternating minimization algorithm is designed to optimize the proposed model, whose computational complexity is linear to the data size. Extensive experiments on ten multi-view datasets, whose data sizes range from 187 to 60 000, have demonstrated the clustering robustness and efficiency of our UDBGL approach over the state-of-the-art MVC approaches.

References

[1] G. Chao, S. Sun, and J. Bi, "A survey on multiview clustering," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 146–168, Apr. 2021.

[2] F. Nie et al., "Self-weighted multiview clustering with multiple graphs," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 2564–2570.

[3] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1261–1270, Mar. 2018.

[4] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, May 2019.

[5] Y. Liang, D. Huang, C.-D. Wang, and P. S. Yu, "Multi-view graph learning by joint modeling of consistency and inconsistency," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 27, 2022, doi: 10.1109/TNNLS.2022.3192445.

[6] W. Liang, X. Liu, S. Zhou, J. Liu, S. Wang, and E. Zhu, "Robust graph-based multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7462–7469.

[7] F. Nie et al., "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 1881–1887.

[8] X.-L. Li, M.-S. Chen, C.-D. Wang, and J.-H. Lai, "Refining graph structure for incomplete multi-view clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 15, 2022, doi: 10.1109/TNNLS.2022.3189763.

[9] Y. Liang, D. Huang, and C.-D. Wang, "Consistency meets inconsistency: A unified graph learning framework for multi-view clustering," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 1204–1209.

[10] H. Gao, F. Nie, X. Li, and H. Huang, "Multi-view subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4238–4246.

[11] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4279–4287.

[12] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Feb. 2011.

[13] G.-Y. Zhang, Y.-R. Zhou, X.-Y. He, C.-D. Wang, and D. Huang, "One-step kernel multi-view subspace clustering," *Knowl.-Based Syst.*, vol. 189, Feb. 2020, Art. no. 105126.

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

[15] Z. Kang et al., "Partition level multiview subspace clustering," *Neural Netw.*, vol. 122, pp. 279–288, Feb. 2020.

[16] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 586–594.

[17] M.-S. Chen, L. Huang, C.-D. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 3513–3520.

[18] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 1–7.

[19] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 1–10.

[20] J. Guo and J. Ye, "Anchors bring ease: An embarrassingly simple approach to partial multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 118–125.

[21] L. Li and H. He, "Bipartite graph based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3111–3125, Jul. 2020.

[22] X. Li, H. Zhang, R. Wang, and F. Nie, "Multiview clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 330–344, Jan. 2022.

[23] Z. Kang, Z. Lin, X. Zhu, and W. Xu, "Structured graph learning for scalable subspace clustering: From single view to multiview," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8976–8986, Sep. 2022.

[24] F. Nie, W. Chang, R. Wang, and X. Li, "Learning an optimal bipartite graph for subspace clustering via constrained Laplacian rank," *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 1235–1247, Feb. 2023.

[25] M. Sun et al., "Scalable multi-view subspace clustering with unified anchors," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*, 2021, pp. 3528–3536.

[26] S. Wang et al., "Fast parameter-free multi-view subspace clustering with consensus anchor guidance," *IEEE Trans. Image Process.*, vol. 31, pp. 556–568, 2021.

[27] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 4412–4419.

[28] X. Lu and S. Feng, "Structure diversity-induced anchor graph fusion for multi-view clustering," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 2, pp. 1–18, 2022.

[29] D. Huang, C.-D. Wang, J.-S. Wu, J.-H. Lai, and C.-K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1212–1226, Jun. 2020.

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.

[30] D. Huang, C.-D. Wang, and J.-H. Lai, "Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 13, 2023, doi: 10.1109/TKDE.2023.3236698.

[31] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[32] S. Huang, I. Tsang, Z. Xu, J. Lv, and Q.-H. Liu, "Multi-view clustering on topological manifold," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 6, pp. 6944–6951.

[33] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010.

[34] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Jul. 2007.

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

[36] J. Huang, F. Nie, and H. Huang, "A new simplex sparse learning model to measure data similarity for clustering," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2015.

[37] D. P. Bertsekas, "Constrained optimization and Lagrange multiplier methods," in *Computer Science and Applied Mathematics*. New York, NY, USA: Academic, 1982.

[38] X. Cai, D. Huang, G.-Y. Zhang, and C.-D. Wang, "Seeking commonness and inconsistencies: A jointly smoothed approach to multi-view subspace clustering," *Inf. Fusion*, vol. 91, pp. 364–375, Mar. 2023.

[39] Z. Hu, F. Nie, R. Wang, and X. Li, "Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding," *Inf. Fusion*, vol. 55, pp. 251–259, Mar. 2020.

[40] X. Liu et al., "One pass late fusion multi-view clustering," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 6850–6859.

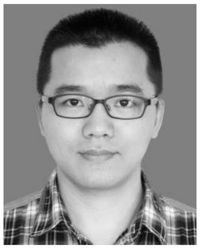
[41] D. Huang, C.-D. Wang, H. Peng, J. Lai, and C.-K. Kwoh, "Enhanced ensemble clustering via fast propagation of cluster-wise similarities," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 508–520, Jan. 2021.

[42] X. Deng, D. Huang, D.-H. Chen, C.-D. Wang, and J.-H. Lai, "Strongly augmented contrastive clustering," *Pattern Recognit.*, vol. 139, Jul. 2023, Art. no. 109470.

[43] S. Liu et al., "Efficient one-pass multi-view subspace clustering with consensus anchors," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 7, pp. 7576–7584.

Si-Guo Fang received the B.S. degree in information and computing sciences from China Jiliang University, Hangzhou, China, in 2021. He is currently pursuing the master's degree in computer science with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China.

His research interests include multi-view clustering and large-scale clustering.



Dong Huang (Member, IEEE) received the B.S. degree in computer science from the South China University of Technology, Guangzhou, China, in 2009, and the M.Sc. and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, in 2011 and 2015, respectively.

He joined South China Agricultural University, Guangzhou, in 2015, where he is currently an Associate Professor with the College of Mathematics and Informatics. From July 2017 to July 2018, he was a Visiting Fellow with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has published more than 70 papers in international journals and conferences, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, *ACM Transactions on Knowledge Discovery From Data*, SIGKDD, AAAI, and ICDM. His research interests include data mining and machine learning.

Dr. Huang was a recipient of the 2020 ACM Guangzhou Rising Star Award.

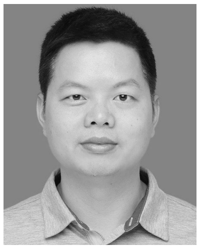
Xiao-Sha Cai received the B.S. degree in software engineering and the M.Sc. degree in computer science from South China Agricultural University, Guangzhou, China, in 2019 and 2022, respectively. She is currently pursuing the Ph.D. degree with Sun Yat-sen University, Guangzhou.

Her research interests include data clustering and information fusion.

Chang-Dong Wang received the B.S. degree in applied mathematics and the M.Sc. and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2008, 2010, and 2013, respectively.

He was a visiting student at the University of Illinois at Chicago, Chicago, IL, USA, from January 2012 to November 2012. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University. He has published more than 100 scientific papers in international journals and conferences such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C, *ACM Transactions on Knowledge Discovery from Data, Pattern Recognition*, SIGKDD, ICDM, and SDM. His current research interests include machine learning and data mining.

Dr. Wang has won the Honorable Mention for Best Research Paper Award for ICDM 2010 paper. He was awarded 2015 Chinese Association for Artificial Intelligence (CAAI) Outstanding Dissertation.



Chaobo He received the B.S., M.S., and Ph.D. degrees from South China Normal University, Guangzhou, China, in 2004, 2007, and 2014, respectively.

He is currently a Professor with the School of Computer Science, South China Normal University. He has published over 30 papers in international journals and conferences. His research interests are data mining and social computing.

Yong Tang received the B.S. and M.Sc. degrees from Wuhan University, Wuhan, China, in 1985 and 1990, respectively, and the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2001, all in computer science.

In 2009, he was the Vice Dean with School of Information of Science and Technology, Sun Yat-Sen University. He is currently a Professor and the Dean with the School of Computer Science, South China Normal University, Guangzhou, China. He is the Founder of SCHOLAT, a kind of scholar social network. He has published more than 200 papers and books. He has supervised more than 40 Ph.D. students since 2003, and more than 100 master's students since 1996. His main research areas include data and knowledge engineering, social networking and collaborative computing.

Dr. Tang serves as the Director of Technical Committee on Collaborative Computing of China Computer Federation (CCF) and the Executive Vice President of Guangdong Computer Academy. For more information, visit https://scholat.com/ytang.

Authorized licensed use limited to: Shanxi University. Downloaded on August 14,2023 at 08:55:36 UTC from IEEE Xplore. Restrictions apply.