Information Fusion 100 (2023) 101941

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Cross-view Graph Matching Guided Anchor Alignment for Incomplete Multi-view Clustering

Xingfeng Li [a], Yinghui Sun [a], Quansen Sun [a,*], Zhenwen Ren [b,c,*], Yuan Sun [d]

[a] Department of Computer Science, Nanjing University of Science and Technology, Nanjing, 210094, China
[b] School of National Defence Science and Technology, Southwest University of Science and Technology, Mianyang, 621010, China
[c] Guangxi Key Laboratory of Machine Vision and Intelligent Control, Wuzhou University, Wuzhou, 543002, China
[d] College of Computer Science, Sichuan University, Chengdu, 610044, China

ARTICLE INFO

Keywords:
Incomplete multi-view clustering
Bipartite graph learning
Anchor alignment
Low-rank tensor

ABSTRACT

Multi-view bipartite graph clustering methods select a few representative anchors and then establish a connection with original samples to generate the bipartite graphs for clustering, which maintains impressive performance while reducing time and space complexities. Despite their effectiveness in large-scale applications, few of them focus on cross-view anchor misalignment (CAM) problem. Then, misaligned anchor sets could destroy the consistent graph semantic structure of bipartite graphs across different views, thus hindering subsequent graph fusion and degrading the clustering performance. Especially when it comes to incomplete data, solving CAM problem becomes an intractable challenge. To address this challenge, we propose a novel **Cross-view Graph Matching guided Anchor Alignment (CGMAA)** for incomplete multi-view bipartite clustering. Specifically, we first propose a novel CGMAA framework to address CAM problem by predefining an anchor graph according to the prior anchor information. In addition, we unify CGMAA and bipartite graph tensor learning for incomplete multi-view clustering. Extensive experiments on ten complete/incomplete benchmark datasets demonstrate the effectiveness, efficiency, and superiority of the proposed CGMAA framework.

1. Introduction

Multi-view graph clustering has attracted intensive attention by optimally integrating heterogeneous and homogeneous properties to group unlabeled multi-view data into different clusters [1–4]. By assuming that all views are complete, massive multi-view graph clustering methods [5–9] have been presented due to their validity of capturing the paired similarities between samples and views. Nevertheless, the existing methods usually involve quadratic or cubic complexities with respect to sample number n, hindering the existing mainstream application of large-scale clustering task [10–12].

To effectively alleviate the large-scale problem, a bipartite graph strategy is proposed to merely select m ($m \ll n$) anchors from different views to approximatively describe the whole dataset [13]. Therefore, both the computational and memory complexity can be reduced into linear for the number of samples since the graph size reduces from $n \times n$ affinity graph to $m \times n$ bipartite graph [14,15]. To achieve this goal, some scholars [16,17] commonly select anchors by using the k-means or random sampling strategies. For instance, [16] first leverages the centers of k-means on each view as anchors to separately generate the bipartite graphs. However, k-means and random sampling may cause inferior representational anchors. To select the high representational ability of anchors, [17] proposes a directly alternate sampling strategy for bipartite graph clustering. Although the above methods can achieve large-scale multi-view bipartite graph clustering, the flexibility and effectiveness of anchor sets are greatly limited, since the discarded unselected samples also contain a lot of useful information. Besides, the anchor sets selection and bipartite graph learning are separate, lacking mutual negotiation. Recently, some methods [14,18] attempt to learn anchors rather than sampling, demonstrating the validity of learning flexible anchors. Concretely, all of them learn a consensus anchor matrix to exploit the homogeneous property for multi-view clustering. In fact, different view data may collect from different sensors or different sources. For instance, the three views of the same Web page or news can be originated from text, images, and videos. The heterogeneous property between these multi-view data can cause the different distribution properties, such that forced learning of consensus anchor matrix would cause partial view failure or not full effectiveness.

Most recently, to better capture the heterogeneous property hidden in data, multi-view anchor-correspondence clustering (FMVACC) is the

* Corresponding authors.
E-mail addresses: sunquansen@njust.edu.cn (Q. Sun), rzw@njust.edu.cn (Z. Ren).

https://doi.org/10.1016/j.inffus.2023.101941
Received 20 February 2023; Received in revised form 10 June 2023; Accepted 19 July 2023
Available online 26 July 2023

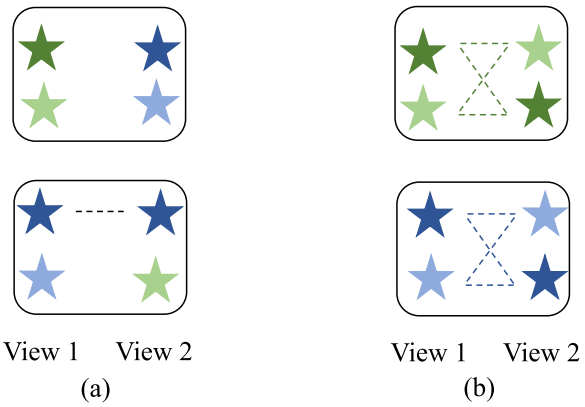


Fig. 1. Motivation of our CGMAA. Fig. 1 takes two views as an example, each view has two classes, where each class contains two anchors. Different colors denote different categories of the learned virtual anchors. The dotted line represents the desired correlation between cross-view anchors. (a) cross-view anchor-level misalignment (CAM): as showcased in Fig. 1(a), the number and order of cross-view anchors are not aligned. For one thing, view 1 learns two blue anchors while view 2 learns three blue anchors, so the anchor number of the blue cluster is not aligned for two views. For another, both view 1 and view 2 learn a wathet blue star, but its order arranges in the second row of view 1 while in the fourth row of view 2, such that the anchor order of the green cluster is not aligned for two views. (b) cross-view cluster-level alignment (CCA): as showcased in Fig. 1(b), each pair is composed of the virtual anchors belonging to the same class. The problem of CAM can be transformed into easier CCA.

first study to learn view-specific anchors from the different views for multi-view bipartite clustering. Although FMVACC has achieved promising improvement, it still suffers from the challenging problem as mentioned in Fig. 1(a). Concretely, view 1 respectively learns 2 green anchors and 2 blue anchors in two different clusters, while for view 2, 3 green anchors and 1 blue anchor are obtained in two different clusters. Note that using these two-view anchors to build the bipartite graphs for subsequent graph fusion is not reasonable, since the graph structure of two bipartite graphs is inconsistent with each other. In this paper, we call this problem a cross-view anchor-level misalignment (CAM) problem. Overall, FMVACC cannot handle the anchor number misalignment problem of CAM, moreover, it only works when the view is complete. Compared to complete multi-view data clustering, a random sample missing not simply results in the loss of view information, but also destroys the paired similarity between views and samples. Thus, the CAM problem will become a trickier challenge for incomplete multi-view data clustering due to the influence of missing information. In this paper, we attempt to address the proposed CAM problem for incomplete multi-view bipartite graph clustering. Specifically, we solve the CAM problem by enforcing anchors of the same cluster to be similar. By this way, the problem of CAM in Fig. 1(a) can be transformed into an easier cross-view cluster-level alignment problem in Fig. 1(b). The detailed illustration is provided in Fig. 1.

To achieve cross-view cluster-level alignment, in this paper, we propose **Cross-view Graph Matching guided Anchor Alignment (CGMAA)** framework to achieve the cross-view cluster-level alignment for incomplete multi-view bipartite clustering. To our knowledge, this paper is the first study of the CAM problem at the cluster level, more importantly, we attempt to address the CAM problem for incomplete multi-view data, further greatly increasing the challenge of CAM. As mentioned in Fig. 2, we first predefine an anchor graph matrix, which can depict and determine the similarity between two bipartite graphs according to the prior information of anchor number choice for each cluster. Then, we use this predefined anchor graph to form the CGMAA framework by directly achieving the column-wisely matching between cross-view bipartite graphs. Meanwhile, graph column-wisely matching can indirectly supervise the learned anchors to achieve the cross-view cluster-level alignment. Finally, by feeding into incomplete multi-view data, CGMAA framework and a high-order bipartite graph tensor learning are jointly performed. After that, multiple bipartite graphs are obtained to generate a fusion graph, whose left singular vector is fed into k-means to get the final clustering result. Main contributions are as follows.

- We observe a significant CAM problem, and transform this tricky problem from CAM to an easier CCA problem. The study of CAM is a pioneering work for multi-view bipartite graph clustering and further benefits the research community.
- We propose a novel Cross-view Graph Matching guided Anchor Alignment (CGMAA) framework to address the proposed CCA problem.
- We employ the proposed CGMAA framework to perform incomplete multi-view bipartite graph clustering, where the superiority and efficiency are all verified compared to the existing advances.

2. Related work

In this section, we introduce the existing work most related to our proposed method, including Incomplete Multi-view Clustering (IMVC) and bipartite graph clustering. The main used notations for the whole paper are listed in Table 1.

2.1. Incomplete multi-view clustering

Recently, IMVC has attracted intensive attention since multimodal data collected from real applications tend to be inherently incomplete, where incompleteness leads to severe performance degradation and even execution failure. Consequently, how to cluster with multi-view partial data becomes a challenging and valuable issue. To effectively alleviate this issue, a series of pioneer works have been proposed, and are able to roughly fall into three classes. For the first category, some scholars [19–23] employ matrix factorization to synergistically generate a consistent representation matrix among incomplete multi-view data. After that, the observed data from different views can be aggregated into a single representation, making it possible to describe the information from multiple incomplete views. Then, consensus representation is put into k-means operation to gain the ultimate results. Secondly, different from sharing consensus representation, kernel-based IMVC methods [24–28] perform IMVC by synergistically learning a consistent kernel or consistent partition across incomplete views to discover the nonlinear information of incomplete views. For the third class, other scholars employ different graph learning techniques to synergistically generate a consensus graph across multiple incomplete views [26, 29–31]. Such as [31,32] use self-representation subspace learning or adaptive neighborhood graph learning to generate a shared similarity graph for spectral clustering. In addition, [31,33] use incomplete multi-view data to construct the graph tensor to explore the high-order correlations hidden in the incomplete data, achieving promising performance. Although effective, the cubic computational complexity and quadratic memory complexity have limited their application for large-scale tasks.

2.2. Bipartite graph clustering

Bipartite graph learning is deemed a very effective strategy to handle large-scale data by selecting a relatively small proportion of representative anchors to establish a connection with original samples. This idea of a multi-view framework can be traditionally expressed as

$$\min_{\mathbf{Z}^p} \|\mathbf{X}^p - \mathbf{Z}^p\mathbf{B}^p\|_{\mathbf{F}}^2 \ \text{s.t.}\ \mathbf{Z}^p \geq 0, \mathbf{Z}^p\mathbf{1} = \mathbf{1} \tag{1}$$

where $\mathbf{X}^p \in \mathbb{R}^{n\times d^p}$ and $\mathbf{B}^p \in \mathbb{R}^{m\times d^p}$ represent complete data and its m selected or sampled anchors corresponding to pth view [16,17]. Eq. (1) can reduce both computational and space complexity since the size of traditional $n \times n$ similarity graph is decreased to $n \times m$ bipartite graph

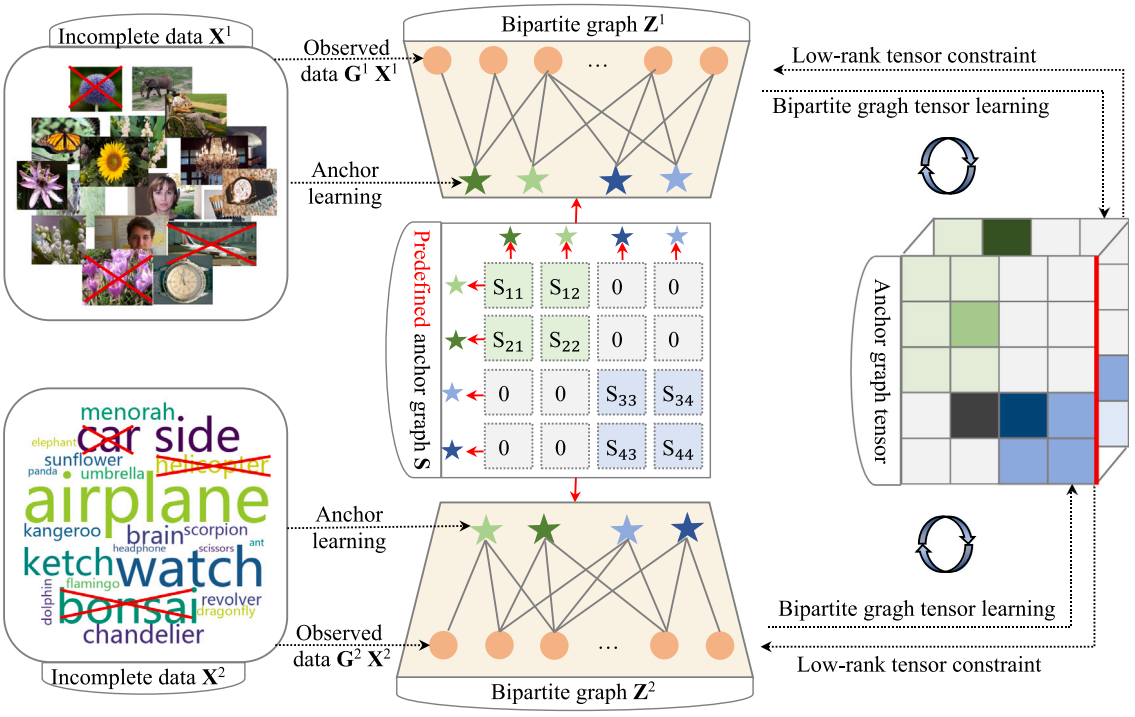


Fig. 2. Overview of the proposed CGMAA method. Two views are employed for ease of understanding. From the left part, original incomplete data $\mathbf{X}^p$ of two views are fed into the middle bipartite graph learning, where red 'cross' denotes the missing samples and $\mathbf{G}^p\mathbf{X}^p$ denotes the observed data of $\mathbf{X}^p$. For the middle part, the predefined anchor graph $\mathbf{S}$ guides the learning of the cross-view anchors and bipartite graphs, and then these bipartite graphs are fed into the bipartite graph tensor learning of the right part.

Table 1
Detailed information of notations.

Notations	Definitions
$\mathbf{X}^p \in \mathbb{R}^{d^p \times n}$	The pth incomplete multi-view data matrix
$\mathbf{B}^p \in \mathbb{R}^{m \times d^p}$	The pth selected anchor matrix in original space
$\mathbf{P}^p \in \mathbb{R}^{m \times k}$	The pth learned anchor matrix in the latent space
$\mathbf{Z}^p \in \mathbb{R}^{m \times n}$	The pth bipartite graph matrix
$\mathcal{Z} \in \mathbb{R}^{V \times m \times n}$	Bipartite graph tensor
$\mathbf{G}^p \in \mathbb{R}^{n \times n_p}$	The pth indicator matrix for missing data
$\mathbf{S} \in \mathbb{R}^{m \times m}$	A predefined anchor graph according to the prior information of anchor setting.
n	The number of samples
d_p	The dimension of samples for pth view
c	The number of clusters
k	The dimension of latent space

$\mathbf{Z}^p \in \mathbb{R}^{m \times n}$. A bipartite graph $\mathbf{Z}^p$ can characterize the relationship between the anchors and the original data. Considering that k-means or sampling may destroy the flexibility of anchors, some methods [14,18] attempt to learn anchors and bipartite graphs in a unified framework. Then, [34] proposes a graph matching framework to address the order misalignment problem of bipartite graphs. However, [34] cannot address our observed CAM problem. We will address this CAM problem in the next section for more difficult incomplete data.

3. Formulation

This section proposes the Cross-view Graph Matching guided Anchor Alignment (CGMAA) framework to handle the cross-view anchor misalignment (CAM) problem proposed in Fig. 1(a), then we employ this framework for incomplete multi-view bipartite graph clustering.

3.1. Learning model of CGMAA

Given d^p dimensions and n samples of complete multi-view data $\mathbf{X}^p$ for multi-view bipartite graph clustering, although learning anchors can enhance the flexibility and effectiveness compared to selecting or sampling the fixed anchors in Eq. (1), these learned anchors commonly suffer from the CAM problem as mentioned in Fig. 1(a). CAM problem could cause low-quality fused bipartite graphs to further reduce the clustering performance. In general, the main challenges of handling CAM problem include two aspects: (1) For each cluster of different views, **how to ensure the learned number of anchors equal;** we take the blue category of Fig. 1(a) as an example, view 1 learns two blue anchors while view 2 learns three blue anchors in the same cluster, which results in an inconsistent bipartite graph structure, hindering subsequent graph fusion; (2) Assuming the anchor number of each category is equal, for different views, **how to make the anchors of same category enjoy consistent order correspondence;** in Fig. 1(a), the wathet blue star arranges in a second order for view 1 while it arranges in fourth order for view 2. Both problems (1) and (2) could greatly decrease the final clustering performance.

To address this tricky CAM problem, we transform this problem from Fig. 1(a) to an easier cross-view cluster-level alignment (CCA) problem of Fig. 1(b). The basic idea of achieving CCA is shown in Fig. 3. From Fig. 3, we predefine a $m \times m$ anchor graph matrix $\mathbf{S} \in \mathbb{R}^{m \times m}$ to reduce the cross-cluster similarity (red dotted line) between two views while enlarging the intra-cluster similarity (green and blue dotted lines) between two views. In the ideal case, the edges of full-connection anchor similarity can be decreased to the edges of ideal

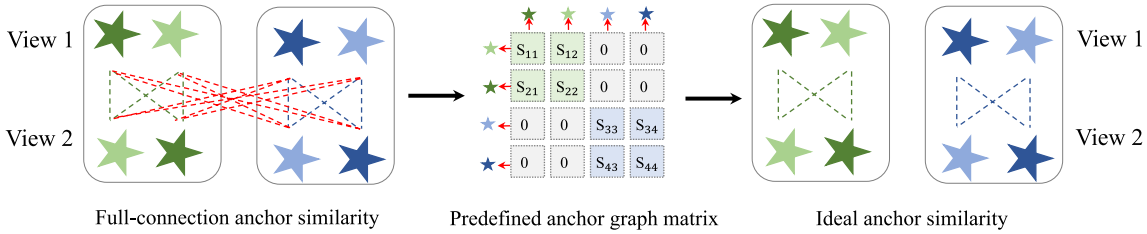


Fig. 3. The basic idea of achieving CCA: A prior anchor graph matrix is predefined to remove the inter-cluster anchor similarity (red dotted lines) of full-connection anchor similarity across views, while enlarging the intra-cluster anchor similarity. The ultimate goal is to obtain an ideal anchor similarity.

anchor similarity with the help of the predefined $\mathbf{S}$. Concretely, we use the predefined anchor graph to construct the prior cross-view bipartite graph matching term. Unlike selecting or sampling the fixed anchors $\mathbf{B}^p \in \mathbb{R}^{m\times d^p}$ of Eq. (1) in the original space, we impose this prior term on the Eq. (1) to dynamically learn anchors $\mathbf{P}^p \in \mathbb{R}^{m\times k}$ in the latent embedding spaces. The above idea can be formalized into

$$\begin{aligned}\min_{\mathbf{W}^p,\mathbf{P}^p,\mathbf{Z}^p} &\sum_{p=1}^{v}\|\mathbf{X}^p\mathbf{W}^p-\mathbf{Z}^p\mathbf{P}^p\|_F^2+\beta\sum_{p=1}^{v}\sum_{r\neq p,r=1}^{v}\sum_{i=1}^{m}\sum_{j=1}^{m}\|\mathbf{z}_i^r-\mathbf{z}_j^p\|_F^2\mathbf{S}_{ij}\\ &\text{s.t. } \mathbf{Z}^p\geq 0,\mathbf{Z}^p\mathbf{1}=\mathbf{1},(\mathbf{W}^p)^\intercal\mathbf{W}^p=\mathbf{I}_k,\mathbf{P}^p(\mathbf{P}^p)^\intercal=\mathbf{I}_m\end{aligned} \tag{2}$$

where $\mathbf{W}^p \in \mathbb{R}^{d^p\times k}$ is the projection matrix with orthogonal constraint, and β is a parameter to control the influence of the prior cross-view bipartite graph matching term. Anchor matrices are imposed on orthogonal constraints to make the anchors more diverse and discriminative. Note herein that the predefined anchor graph is denoted as

$$\mathbf{S}_{ij}=\begin{cases}\frac{1}{n_m}, & \text{if } \mathbf{p}_i^p \text{ and } \mathbf{p}_j^r \text{ in same cluster } (p\neq r)\\ 0, & \text{otherwise.}\end{cases} \tag{3}$$

where n_m is the predefined anchor number of each cluster. $\frac{1}{n_m}$ encodes the desired similarity between the same clusters of two cross-view anchors. That is, if an anchor $\mathbf{p}_i^p$ in the pth view is in the same cluster with an anchor in rth view, the similarity between these two anchors should be $\frac{1}{n_m}$. Inversely, if these two anchors are in different clusters, their similarities should be zeros. Taking the data with two views and two clusters as an example, all the zeros in $\mathbf{S}$ encode the similarities of different clusters of two views, while non-zero values of $\mathbf{S}$ encode the similarities of same clusters. According to similarities with nonzero and zero of anchor prior information, we can predefine the ideal predefined anchor graph $\mathbf{S}$ of Fig. 3. In fact, $\frac{1}{n_m}$ is adjustable, but for convenience, the anchor number of each cluster is fixed to be equal. By this way, Eq. (2) can indirectly achieve the ideal anchor similarity via the prior cross-view bipartite graph matching term. The reason is that the closer cross-view anchors in the latent feature space have the closer cross-view bipartite graph representations in the graph semantic space, *i.e.*, the smaller $\|\mathbf{p}_i^p-\mathbf{p}_j^r\|_F$ can promote to the smaller $\|\mathbf{z}_i^p-\mathbf{z}_j^r\|_F$, vice versa. Inspired by this, according to the predefined anchor graph $\mathbf{S}$, the predefined prior cross-view matching term can directly make the intra-cluster two columns of two cross-view bipartite graphs become closer, meanwhile making inter-cluster two columns become bigger. Additionally, this predefined matching term can indirectly make intra-cluster rows of the learned anchor matrices across different views become closer, meanwhile making inter-cluster anchors become bigger. For the ideal case, only the intra-cluster anchors have bigger affinity values than zero while the affinity values between inter-cluster anchors are all zeros as mentioned in the right of Fig. 3. By this way, Eq. (2) can technically handle the proposed CAM problem.

3.2. CGMAA for incomplete multi-view bipartite graph clustering

With the CGMAA framework at hand, we employ it to perform incomplete multi-view bipartite graph clustering. First, we introduce the index matrices $\{\mathbf{G}^p\}_{p=1}^v \in \mathbb{R}^{n_p\times n}$ into Eq. (2) to prevent the negative impact of missing samples on anchor learning. n_p is the number of observed data of pth view. Then, a t-SVD based tensor low-rank constraint is imposed on bipartite graphs to deeply investigate the high-order correlations of data. So far, the ultimate objective function can be mathematically fulfilled as

$$\begin{aligned}\min_{\mathbf{P}^p,\mathbf{W}^p,\mathbf{Z}^p,\mathcal{Z}} &\sum_{p=1}^{v}\|\mathbf{G}^p\mathbf{X}^p\mathbf{W}^p-\mathbf{G}^p\mathbf{Z}^p\mathbf{P}^p\|_F^2\\ &+\beta\sum_{p=1}^{v}\sum_{r\neq p,r=1}^{v}\sum_{i=1}^{m}\sum_{j=1}^{m}\|\mathbf{z}_i^r-\mathbf{z}_j^p\|_F^2\mathbf{S}_{ij}+\gamma\|\mathcal{Z}\|_{\circledast}\\ &\text{s.t. } \mathbf{Z}^p\geq 0,\mathbf{Z}^p\mathbf{1}=\mathbf{1},(\mathbf{W}^p)^\intercal\mathbf{W}^p=\mathbf{I}_k,\mathbf{P}^p(\mathbf{P}^p)^\intercal=\mathbf{I}_m\end{aligned} \tag{4}$$

where the function $\Psi(\cdot)$ stacks p bipartite graph into a third-order tensor, *i.e.*, $\mathcal{Z}=\Psi(\mathbf{Z}^1,\ldots,\mathbf{Z}^m)$. γ is a regularization parameter. And $\mathbf{G}_{ij}^p$ is pth index matrix and defined as

$$\mathbf{G}_{ij}^p=\begin{cases}1, & \text{if the entry } g_{i,j}^p=i,\ \forall j=1,2,\ldots,n^p\\ 0, & \text{otherwise.}\end{cases} \tag{5}$$

where $\mathbf{g}^p$ is the indicator vector of n_p observed samples to report the sorted index. $g_{i,j}^p$ denotes ith column and jth row element of $\mathbf{g}^p$ for pth view. $\{\mathbf{G}^p\mathbf{X}^p\in\mathbb{R}^{n^p\times d^p}\}_{p=1}^v$ are the sorted observed samples of v views, where n^p is the observed sample number.

3.3. Optimization

In this subsection, we can observe that the alternating direction method of multipliers (ADMM) can be used to solve Eq. (4) since it is convex. We first introduce an auxiliary variable $\mathcal{H}$ to make Eq. (4) separable according to the principle of ADMM [35,36]. Further, by denoting $\otimes$ be dot product between two matrices, we find that $(\mathbf{G}^p)^\intercal\mathbf{G}^p\mathbf{X}^p=\mathbf{A}^p\otimes\mathbf{X}^p$, where $\mathbf{A}^p=\mathbf{g}^p\mathbf{1}_{d^p}$ and $(\mathbf{g}^p)^\intercal=[g_1^p,\ldots,g_n^p]$ with $g_j^p=\sum_{l=1}^{n^p}\mathbf{G}_{j,l}^p$, which can make $\mathcal{O}(vn^2)$ space complexity reduce to $\mathcal{O}(dn)$. Finally, the augmented Lagrangian function of Eq. (4) is

$$\begin{aligned}\min_{\substack{\mathbf{P}^p,\mathbf{W}^p,\\ \mathbf{Z}^p,\mathcal{Y},\mathcal{H}}} &\sum_{p=1}^{v}\|\mathbf{G}^p\mathbf{X}^p\mathbf{W}^p-\mathbf{G}^p\mathbf{Z}^p\mathbf{P}^p\|_F^2+\gamma\|\mathcal{H}\|_{\circledast}\\ &+\beta\sum_{p=1}^{v}\mathrm{Tr}\Big(\sum_{r=1,r\neq p}^{v}(\mathbf{Z}^r)^\intercal\mathbf{L}_S\mathbf{Z}^p\Big)+\frac{\mu}{2}\|\mathcal{Z}-\mathcal{H}+\frac{\mathcal{Y}}{\mu}\|_F^2\\ &\text{s.t. } \mathbf{Z}^p\geq 0,\mathbf{Z}^p\mathbf{1}=\mathbf{1},(\mathbf{W}^p)^\intercal\mathbf{W}^p=\mathbf{I}_k,\mathbf{P}^p(\mathbf{P}^p)^\intercal=\mathbf{I}_m\end{aligned} \tag{6}$$

where Laplacian matrix $\mathbf{L}_S=\mathbf{T}-\mathbf{S}$, and the degree matrix $\mathbf{T}$ is a diagonal matrix with $T_{ii}=\sum_{j=1}\mathbf{S}_{ij}$. Eq. (6) can be solved separately as follows.

▶ **Step-1 update Z**: Ignoring the irrelevant items *w.r.t* $\mathbf{Z}$ in Eq. (6), updating $\mathbf{Z}$ changes to

By eliminating the irrelevant variables of $\mathbf{Z}$ involved in Eq. (6), $\mathbf{Z}$-subproblem becomes to

$$\min_{\mathbf{Z}^p}\ \|\mathbf{Z}^p-\hat{\mathbf{Z}}^p\|_F^2 \text{s.t. } \mathbf{Z}^p\geq 0,\mathbf{Z}^p\mathbf{1}=\mathbf{1}, \tag{7}$$

where $\hat{\mathbf{Z}}^p=\frac{(\mathbf{P}^p)^\intercal(\mathbf{W}^p)^\intercal(\mathbf{A}^p\otimes\mathbf{X}^p)-0.5\beta\mathbf{D}^p+0.5\mu\mathbf{H}^p-0.5\mathbf{Y}^p}{(1+0.5\mu)\mathbf{I}}$ and $\mathbf{D}^p=\mathbf{L}_S^\intercal\sum_{r=1,r\neq p}^{v}\mathbf{Z}^r$. The jth column vector of $\hat{\mathbf{Z}}^p$ is defined as $\hat{\mathbf{z}}_j^p$, whose ith element is $\hat{z}_{i,j}^p$. For each view, we can update $\mathbf{Z}^p$ column by column as follows.

$$\min_{\mathbf{z}_j}\left\|\mathbf{z}_j-\hat{\mathbf{z}}_j\right\|_{\mathbf{F}}^2,\ \text{s.t. } \mathbf{z}_j\mathbf{1}=1,\mathbf{z}_{ij}\geq 0 \tag{8}$$

Such a subproblem can be solved by following Theorem 1.

Theorem 1. *For any r vectors $\{\hat{\mathbf{z}}\}_{j=1}^{r}$, a closed-form solution $\mathbf{z}_j^*$ can be achieved as*

$$\mathbf{z}^* = arg\ \min_{\mathbf{z}} \|\mathbf{z}-\hat{\mathbf{z}}\|_F^2,\ s.t.\ \mathbf{z}^\top\mathbf{1}=1, \mathbf{z}\geq 0 \tag{9}$$

whose specific proof can be found in Theorem 2 of [37]. The time complexity of optimizing $\mathbf{Z}$ is $\mathcal{O}(nmd)$ with the close-form solution.

▶ **Step-2 update $\mathbf{P}^p$**: Optimizing $\mathbf{P}$ with the irrelevant variables fixed is equivalent to the following optimization problem

$$\max_{\mathbf{P}^p} \mathrm{Tr}(\mathbf{P}^p(\mathbf{C}^p)^\top), \mathbf{P}^p(\mathbf{P}^p)^\top = \mathbf{I}_m, \tag{10}$$

where $\mathbf{C}^p = \mathbf{Z}^p(\mathbf{A}_p \otimes \mathbf{X}_p)\mathbf{W}^p$. The optimal solution of optimizing $\mathbf{P}^p$ is gained by performing singular value decomposition (SVD) on $\mathbf{C}^p$ [38] with complexity $\mathcal{O}(v(ndk+nmk+m^2k))$ per iteration, where $d = \sum_{p=1}^{v} d^p$.

▶ **Step-3 update $\mathbf{W}$**: Optimizing $\mathbf{W}$ with the irrelevant variables fixed can be written as

$$\max_{\mathbf{W}^p} \mathrm{Tr}((\mathbf{W}^p)^\top\mathbf{E}^p), (\mathbf{W}^p)^\top\mathbf{W}^p = \mathbf{I}_k, \tag{11}$$

where $\mathbf{E}^p = (\mathbf{A}_p \otimes \mathbf{X}_p)^\top\mathbf{P}^p\mathbf{Z}^p$. The optimal solution of optimizing $\mathbf{W}^p$ is gained by performing SVD $\mathbf{E}^p$ with complexity $\mathcal{O}(vd(nm+km+k^2))$ per iteration.

▶ **Step-4 update $\mathcal{H}$**: Ignoring the irrelevant items *w.r.t* $\mathcal{H}$, updating $\mathcal{H}$ can be rewritten as

$$\min_{\mathcal{H}}\ \gamma\|\mathcal{H}\|_{\circledast} + \frac{\mu}{2}\|\mathcal{H} - (\mathcal{Z} + \frac{\mathcal{Y}}{\mu})\|_F^2 \tag{12}$$

Denoting $\mathcal{M} = \mathcal{Z} + \frac{\mathcal{Y}}{\mu}$, $\mathcal{H}$ can be solved via the tensor tubal-shrinkage of the below Theorem 2.

Theorem 2. *Give two 3-order tensor $\mathcal{H} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{M} \in \mathbb{R}^{n_1\times n_2\times n_3}$ with a scalar, the global optimal solution to the following problem*

$$\min_{\mathcal{H}} \rho\|\mathcal{H}\|_{\circledast} + \frac{1}{2}\|\mathcal{H}-\mathcal{M}\|_F^2 \tag{13}$$

is given by the tensor tubal-shrinkage operator.

$$\mathcal{H} = \mathcal{C}_{n_3\rho}(\mathcal{M}) = \mathcal{U} * \mathcal{C}_{n_3\rho}(\mathcal{Z}) * \mathcal{V}^\top \tag{14}$$

where $\mathcal{M} = \mathcal{U} * \mathcal{Z} * \mathcal{V}^\top$ and $\mathcal{C}_{n_3\rho}(\mathcal{Z}) = \mathcal{Z} * \mathcal{Q}$. $\mathcal{Q} \in \mathbb{R}^{n_1\times n_2\times n_3}$ denotes a f-diagonal tensor and each diagonal element of $\mathcal{Q}$ is defined as $\mathcal{Q}_f(i,j,p) = (1 - \frac{n_3\rho}{\mathcal{Z}^p(i,j)})_+$. The complexity of updating $\mathcal{H}$ is $\mathcal{O}\left(vnm\log(n) + v^2nm\right)$.

Updating ADMM variables are written as

$$\begin{aligned} &\mathcal{Y} = \mathcal{Y} + \mu(\mathcal{Z}-\mathcal{H}) \\ &\mu = min(\rho\mu, \mu_{max}) \end{aligned} \tag{15}$$

where $\mu = 1e^{-4}$ and $\mu_{max} = 10^{10}$, and the complexity is $\mathcal{O}(n)$. Algorithm 1 reports the optimization process of Eq. (6), and obj^t is the objective value of Algorithm 1 to check its convergence at the tth iteration.

Algorithm 1 The algorithm of CGMAA

Input: v complete/incomplete multi-view data $\{\mathbf{X}^p\}_{p=1}^{v}$, cluster number c, latent space dimension k, and parameters β and γ.
Initialize $\mathbf{S}$ via Eq. (3), $\mathbf{G}^p$ via Eq. (5), $\mathbf{W}^p = \mathbf{I}_k$, and the others matrices as $\mathbf{0}$.
1: **repeat**
2: Update $\mathbf{Z}^p$ by using Eq. (7);
3: Update $\mathbf{P}^p$, $\mathbf{W}^p$, and $\mathcal{H}$ via Eqs. (10)–(12);
4: Update ADMM variables via Eq. (15);
5: **until** Satisfy $(obj^{(t)} - obj^{(t-1)})/obj^{(t)} \leq 1e-4$.
6: Perform SVD on the averaged graph $\mathbf{Z}^* = \sum_{p=1}^{v}\mathbf{Z}^p/v$ to obtain left singular value matrix $\mathbf{U}_{\mathbf{Z}^*} \in \mathbb{R}^{n\times m}$.
Output: Perform k-means on $\mathbf{U}_{\mathbf{Z}^*}$ to obtain final results.

Table 2
Complexity analysis of SOTA competitors.

Method	Space cost	Time complexity
BSV [39]	vn^2	$\mathcal{O}\left(n^3\right)$
MIC [19]	$vn^2 + nd + nvk + vdk$	$\mathcal{O}\left(n^3 + n^2dk\right)$
DAIMC [20]	$vn^2 + nd + nk + dk$	$\mathcal{O}\left(nd^3 + ndk\right)$
APMC [40]	$nd + vnm + nk$	$\mathcal{O}\left(n^3 + nmd + m^3\right)$
UEAF [23]	$vn^2 + dn + nvk + dk$	$\mathcal{O}\left(n^3 + dk^2\right)$
MKKM-IK [26]	$vn^2 + vnk$	$\mathcal{O}\left(vn^3\right)$
EEIMVC [27]	$vn^2 + vnk + vk^2$	$\mathcal{O}\left(nk^2 + vk^3\right)$
FLSD [22]	$vn^2 + dnk + nk$	$\mathcal{O}\left(nd^2\right)$
UTF [31]	$vn^2 + v(n-n_p)d$	$\mathcal{O}(vn^3 + vn^2log(n) + v^2n^2)$
IMVC-CBG [41]	$mn + (d+m)k$	$\mathcal{O}(ndk + nmd + mdk)$
HCP-IMSC [33]	$vn^2 + v(n-n_p)d + dk$	$\mathcal{O}(vn^3 + v(n-n_p)^3 + kn^2v)$
Ours	$m(n+k) + d(n+m) + 3mnv$	$\mathcal{O}(vnmd + vmn\log(n))$

3.4. Complexity analysis

The computational complexity of Algorithm 1 has been given from **Step-1** to **Step-4**, separately. We first recall the four constants, *i.e.*, sample number n, cluster number c, anchor number m, and latent space dimension k. Then, for each iteration, **Step-1** requires $\mathcal{O}(vnmd)$ to update column by column. **Step-2** and **Step-3** involves matrix multiplication and SVD operation with complexity $\mathcal{O}(v(ndk+nmk+m^2k))$ and $\mathcal{O}(vd(nm+km+k^2))$, respectively. **Step-4** needs $\mathcal{O}\left(vnm\log(n) + v^2nm\right)$ to complete the tensor calculation. And updating ADMM variables require $\mathcal{O}(v)$ complexity. After obtaining $\mathbf{Z}^*$, it costs $\mathcal{O}(nm^2)$ complexity to perform k-means. Thus, Algorithm 1 involves the total complexity as $\mathcal{O}(t(vnmd+v(ndk+nmk+m^2k)+vd(nm+km+k^2)+vnm\log(n)+v^2nm+nm^2))$, which approximates to $\mathcal{O}(vnmd + vmn\log(n))$.

For another, space complexity of Algorithm 1 mainly involves the following matrices $\mathbf{W}^p \in \mathbb{R}^{d^p\times m}$, $\mathbf{X}^p \in \mathbb{R}^{n\times d^p}$, $\mathbf{P}^p \in \mathbb{R}^{m\times k}$, $\mathbf{Z}^p \in \mathbb{R}^{n\times m}$, $\mathcal{H} \in \mathbb{R}^{m\times n\times v}$, $\mathcal{Z} \in \mathbb{R}^{m\times n\times v}$, and $\mathcal{Y} \in \mathbb{R}^{m\times n\times v}$, where $d = \sum_{p=1}^{v} d_p$. Thus, the major involved space complexity of CGMAA is $m(n+k)+d(n+m)+3mnv$. Due to $c < k$, $k \ll n$, $m \ll n$, and $d \ll n$, Algorithm 1 approximates to linear time and space complexities with respect to the number of samples. Table 2 reports the main time and space complexity of all compared methods.

4. Experiments

4.1. Incomplete datasets and experimental setting

Following the approach in [41], we generate incomplete datasets by setting missing ratio ψ from 0.1 to 0.9 with step of 0.1, *i.e.*, $\psi = \frac{n_p}{n} \in [0.1 : 0.1 : 0.9]$. Ten widely used benchmark datasets are employed, including: Cifar10, Cifar100,[1] NUSWIDE, [2] SUNRGBD,[3] UCI-Digit,[4] Caltech-20,[5] NGs,[6] ORL,[7] BDGP,[8] and MNIST.[9] Detailed information on these datasets is provided in Table 3. Concretely, Cifar100, Cifar10, NUSWIDE, SUNRGBD, Caltech-20, BDGP, UCI-Digit, MNIST, and ORL are the image datasets. NGs is the web page dataset. Note that the number of samples in these datasets ranges from 400 to 60,000. This span is already relatively large in existing complete/incomplete multi-view clustering. All evaluated complete/incomplete datasets in this paper are collected from the publicly published in the literatures [31, 34,41].

[1] http://cs.toronto.edu/kriz/cifar.html
[2] https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html
[3] https://rgbd.cs.princeton.edu/
[4] http://archive.ics.uci.edu/ml/datasets/Multiple+Features
[5] http://www.vision.caltech.edu/ImageDatasets/Caltech101/
[6] https://lig-membres.imag.fr/grimal/data.html
[7] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedataset.html
[8] https://www.fruitfly.org/
[9] http://yann.lecun.com/exdb/mnist/

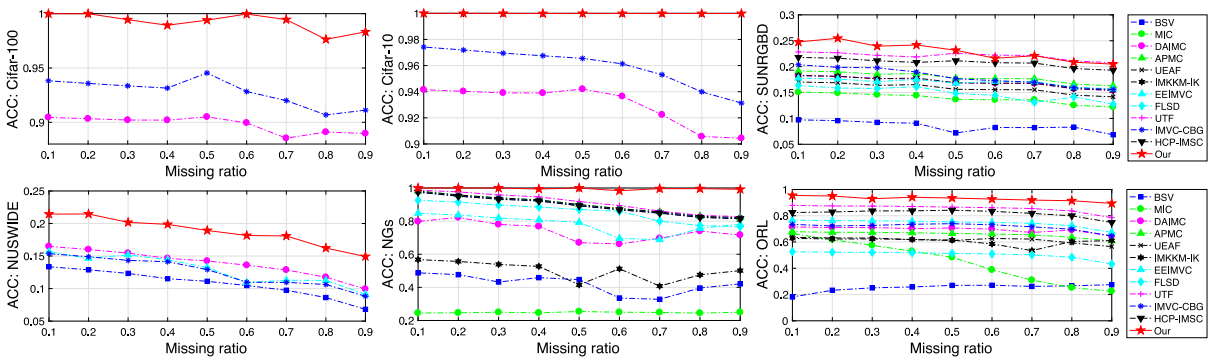


Fig. 4. The averaged ACC variation with missing ratios from 0.1 to 0.9 on six benchmark datasets are reported.

Table 3
The number of samples, classes, views, and dimensionality of the used datasets.

Dataset	Sample	Classes	Views	Dimensionality
ORL	400	40	3	4096 /3304/6750
NGs	500	5	3	500/500/500
Caltech-20	2386	20	6	48/40/254/198/512/928
SUNRGBD	10335	45	2	4096/4096
NUSWIDE	30000	31	5	65/226/145/74/129
Cifar10	50000	10	3	2048/512/1024
Cifar100	50000	100	3	2048/512/1024
UCI-Digit	2000	10	3	240/76/6
BDGP	2500	5	3	1000/500/250
MNIST	60000	10	3	342/64/1024

One baseline and ten state-of-the-art (SOTA) IMVC methods are employed for comparison, including, **BSV** [39] (the optimal value with single view mean imputation), **MIC** [19], **APMC** [40], **FLSD** [22], **DAIMC** [20], **UEAF** [23], **IMKKM-IK** [26], **EEIMVC** [27], **UTF** [31], **HCP-IMSC** [33], **IMVC-CBG** [41], and our **CGMAA**. The computing platform is an Intel Core i9 CPU and 64 GB RAM and codes from their public homepage. Most of the codes come from the pages of Jie Wen[10] and Xinwang Liu.[11]

4.2. Experiment results for incomplete datasets

To demonstrate the superiority of our CGMAA, four common metrics [27] *i.e.*, Fscore (FSC), Purity (PUR), normalized mutual information (NMI), and Accuracy (ACC) are employed to compare with 11 state-of-the-art incomplete multi-view clustering methods on seven incomplete multi-view datasets. The average clustering performance and standard deviations (Each experiment is repeated 20 times.) on all evaluated datasets are reported in Table 4. The best and second-best averages are marked in bold and underlined, respectively. Meanwhile, Figs. 4–7 present the variation for our method with different missing ratios on all evaluated metrics. From these tables and figures, we can observe that:

Effectiveness. Some IMVC methods are worse than single-view method BSV, while our CGMAA not only consistently outperforms the single-view incomplete clustering, but outperforms all the recently proposed IMVC methods in most cases. This demonstrates the effectiveness of our method in IMVC.

Superiority over graph tensor based methods, *i.e.*, UTF and HCP-IMSC, which have been regarded as the strongest graph tensor based IMVC methods. For datasets under 30,000 samples (*i.e.*, ORL, NGs, Caltech-20, and SUNRGBD), except NMI on the Caltech-20 dataset, our method has a significant improvement in all evaluated metrics and also takes a lot less time cost. For datasets from 30,000 to 50,000 samples (*i.e.*, NUSWIDE, Cifar10, and Cifar100), UTF and HCP-IMSC fail due to their quadratic memory complexity and cubic time complexity. Comparatively, even when the missing ratio is 0.9, the performance of our method is close to 100% on the Cifar10 and Cifar100 datasets. Overall, our method outperforms UTF and HCP-IMSC by a large margin, achieving less time cost.

Superiority over anchor learning based method, *i.e.*, IMVC-CBG, which first attempts to joint learn a shared anchor matrix and a consensus bipartite graph for IMVC, obtaining promising results recently. Regrettably, it can only capture the view-consistent information of data, ignoring the view-specific information. Contrastively, our CGMAA can simultaneously capture view-consistent, view-specific, and high-order graph information. Therefore, although both IMVC-CBG and CGMAA have similar low time consumption in Table 7, the clustering results of CGMAA consistently outperform IMVC-CBG on all datasets and metrics.

4.3. Ablation analysis

To demonstrate the significance and effectiveness of cross-view graph matching term, we remove the second cross-view graph matching term of Eq. (4) by setting $\beta = 0$ (*i.e.*, CGMAA$_{\beta=0}$) on all the evaluated datasets and metrics. The results are recorded in Table 5. Further, to make a more visual comparison in Fig. 8, we visualize p embedding representations $\mathbf{P}^p\mathbf{Z}^p$ with their anchors $\mathbf{P}^p$ on the NGs dataset for CGMAA and CGMAA$_{\beta=0}$, respectively. Table 5 indicates that CGMAA enjoys better clustering results with the help of cross-view graph matching term. Meanwhile, Fig. 8 also demonstrates that the cross-view graph matching term can make the anchor number of each cluster of different views equal, and the anchor order of different views is also aligned. Thus, the learned data of CGMAA enjoy the clearer cluster discriminant.

4.4. Experiment results for complete datasets

This section compares the SOTA competitor FMVACC [34] and our CGMAA on the ORL, NUSWIDE, UCI-Digit, BDGP, SUNRGBD, and MNIST datasets. Both FMVACC [34] and our CGMAA are multi-view bipartite graph clustering methods to address bipartite graph matching problems for the first practice. The results reported in Table 6 indicate that CGMAA has a considerable improvement over FMVACC. The main reason is that FMVACC cannot address the intractable CAM problem, while CGMAA can alleviate this problem technically. Concretely, FMVACC can only handle the problem that the anchor number of each cluster for different views is equal. While our CGMAA can simultaneously make the anchor number and anchor order in each cluster align. In fact, the anchor number learned from FMVACC are usually not equal like in Fig. 8(a)–(c), thus resulting in less clustering performance.

[10] https://sites.google.com/view/jerry-wen-hit/publications
[11] https://xinwangliu.github.io/

Table 4
Average clustering comparison of 0.1 to 0.9 missing ratios on the seven datasets. The best results are marked in bold and the second-best results are marked in underline. "-" means out of the storage memory or out of the CPU memory.

Dataset	BSV	MIC	DAIMC	APMC	UEAF	IMKKM-IK	EEIMVC	FLSD	UTF	IMVC-CBG	HCP-IMSC	Our
ACC												
ORL	25.18 ± 0.73	44.94 ± 1.50	69.45 ± 2.16	65.80 ± 1.75	61.49 ± 2.34	60.41 ± 2.73	74.36 ± 2.38	50.33 ± 1.69	85.57 ± 0.16	71.65 ± 2.53	81.88 ± 0.03	**92.95 ± 0.02**
NGs	42.37 ± 1.83	24.86 ± 0.26	73.31 ± 9.07	89.89 ± 0.01	89.89 ± 0.03	50.36 ± 0.13	78.37 ± 0.12	85.47 ± 0.03	91.13 ± 0.09	89.26 ± 0.02	89.11 ± 0.00	**99.57 ± 0.00**
Caltech-20	39.91 ± 0.23	26.69 ± 1.68	45.72 ± 1.93	–	39.71 ± 1.38	32.21 ± 1.66	41.33 ± 1.29	43.45 ± 1.82	49.48 ± 2.09	50.56 ± 1.32	50.05 ± 0.46	**56.38 ± 0.01**
SUNRGBD	8.49 ± 0.06	13.84 ± 0.48	17.01 ± 0.76	17.93 ± 0.55	15.80 ± 0.39	17.06 ± 0.36	16.97 ± 0.46	14.82 ± 0.39	22.04 ± 1.67	18.05 ± 0.18	20.76 ± 0.06	**22.96 ± 0.01**
NUSWIDE	10.76 ± 0.20	–	13.89 ± 0.18	–	–	–	12.86 ± 0.26	–	–	12.56 ± 0.13	–	**18.81 ± 0.01**
Cifar10	–	–	90.81 ± 0.45	–	–	–	–	–	–	96.19 ± 0.13	–	**99.99 ± 0.00**
Cifar100	–	–	89.71 ± 1.00	–	–	–	–	–	–	93.09 ± 1.18	–	**99.24 ± 0.00**
NMI												
ORL	48.92 ± 0.75	57.44 ± 0.84	82.44 ± 0.9	81.10 ± 0.66	77.58 ± 1.09	79.60 ± 1.10	85.93 ± 1.16	68.07 ± 1.12	92.98 ± 0.08	80.61 ± 1.46	90.85 ± 0.02	**93.99 ± 0.01**
NGs	22.79 ± 1.17	9.28 ± 0.29	58.93 ± 6.28	73.73 ± 0.03	73.73 ± 0.08	33.49 ± 0.11	58.92 ± 0.18	66.48 ± 0.03	75.92 ± 0.15	73.90 ± 0.06	72.90 ± 0.00	**98.68 ± 0.00**
Caltech-20	25.58 ± 0.83	30.66 ± 1.13	55.70 ± 1.36	–	50.90 ± 0.92	40.09 ± 0.98	54.74 ± 0.61	52.33 ± 0.90	**72.31 ± 1.32**	52.86 ± 1.61	60.37 ± 0.53	67.62 ± 0.01
SUNRGBD	6.31 ± 0.06	21.02 ± 0.28	21.34 ± 0.35	21.83 ± 0.22	21.21 ± 0.28	20.95 ± 0.28	20.86 ± 0.24	20.86 ± 0.19	35.00 ± 1.32	23.92 ± 0.16	33.72 ± 0.08	**36.52 ± 0.00**
NUSWIDE	3.39 ± 0.06	–	11.34 ± 0.39	–	–	–	10.52 ± 0.09	–	–	10.40 ± 0.03	–	**17.66 ± 0.00**
Cifar10NMI	–	–	90.47 ± 0.55	–	–	–	–	–	–	90.89 ± 0.27	–	**99.96 ± 0.00**
Cifar100	–	–	98.26 ± 0.16	–	–	–	–	–	–	98.63 ± 0.29	–	**99.78 ± 0.00**
PUR												
ORL	26.34 ± 0.76	41.17 ± 1.24	73.83 ± 1.63	70.18 ± 1.32	64.05 ± 1.75	61.56 ± 2.47	76.82 ± 2.03	51.49 ± 1.56	88.13 ± 0.12	71.65 ± 2.02	84.44 ± 0.03	**90.51 ± 0.01**
NGs	44.37 ± 1.31	25.79 ± 0.22	73.31 ± 9.07	89.89 ± 0.01	89.89 ± 0.03	51.23 ± 0.10	78.37 ± 0.12	85.47 ± 0.03	91.13 ± 0.09	89.26 ± 0.02	89.11 ± 0.00	**99.57 ± 0.00**
Caltech-20	48.70 ± 0.69	55.89 ± 1.13	74.89 ± 0.88	–	70.90 ± 0.74	60.14 ± 1.38	72.28 ± 0.79	73.37 ± 0.53	**82.22 ± 1.76**	71.04 ± 0.93	76.55 ± 0.45	73.77 ± 0.01
SUNRGBD	13.22 ± 0.18	32.58 ± 0.64	34.90 ± 0.81	34.99 ± 0.39	34.78 ± 0.62	34.51 ± 0.86	34.42 ± 0.37	32.66 ± 0.34	42.55 ± 1.64	33.55 ± 0.28	41.27 ± 0.03	**44.56 ± 0.01**
NUSWIDE	14.75 ± 0.08	–	23.69 ± 0.32	–	–	–	21.88 ± 0.23	–	–	21.01 ± 0.15	–	**27.61 ± 0.01**
Cifar10	–	–	95.81 ± 0.45	–	–	–	–	–	–	96.19 ± 0.26	–	**99.99 ± 0.00**
Cifar100	–	–	92.60 ± 0.54	–	–	–	–	–	–	94.98 ± 0.82	–	**99.26 ± 0.00**
FSC												
ORL	9.86 ± 0.53	18.80 ± 1.02	56.97 ± 2.71	50.73 ± 2.39	40.23 ± 2.58	45.20 ± 2.91	64.52 ± 2.69	31.91 ± 1.84	79.10 ± 0.06	47.05 ± 3.83	75.42 ± 0.01	**91.42 ± 0.02**
NGs	32.97 ± 0.88	33.27 ± 0.13	74.11 ± 7.29	81.29 ± 0.02	81.29 ± 0.06	42.84 ± 0.09	64.00 ± 0.14	72.02 ± 0.03	80.04 ± 0.11	81.23 ± 0.03	80.04 ± 0.00	**99.15 ± 0.00**
Caltech-20	32.74 ± 0.29	24.46 ± 1.26	40.46 ± 2.19	–	33.34 ± 1.49	29.18 ± 1.65	40.70 ± 1.43	41.50 ± 1.56	43.56 ± 2.62	44.59 ± 2.32	44.81 ± 0.68	**54.67 ± 0.01**
SUNRGBD	7.18 ± 0.02	9.53 ± 0.18	10.66 ± 0.32	10.98 ± 0.28	9.50 ± 0.14	10.11 ± 0.19	10.02 ± 0.22	11.45 ± 0.01	16.16 ± 1.19	10.51 ± 0.38	14.88 ± 0.05	**16.38 ± 0.00**
NUSWIDE	**10.75 ± 0.05**	–	8.62 ± 0.59	–	–	–	8.01 ± 0.06	–	–	7.91 ± 0.03	–	10.37 ± 0.00
Cifar10	–	–	92.16 ± 0.68	–	–	–	–	–	–	92.67 ± 0.13	–	**99.98 ± 0.00**
Cifar100	–	–	90.82 ± 0.94	–	–	–	–	–	–	90.87 ± 2.97	–	**99.16 ± 0.00**

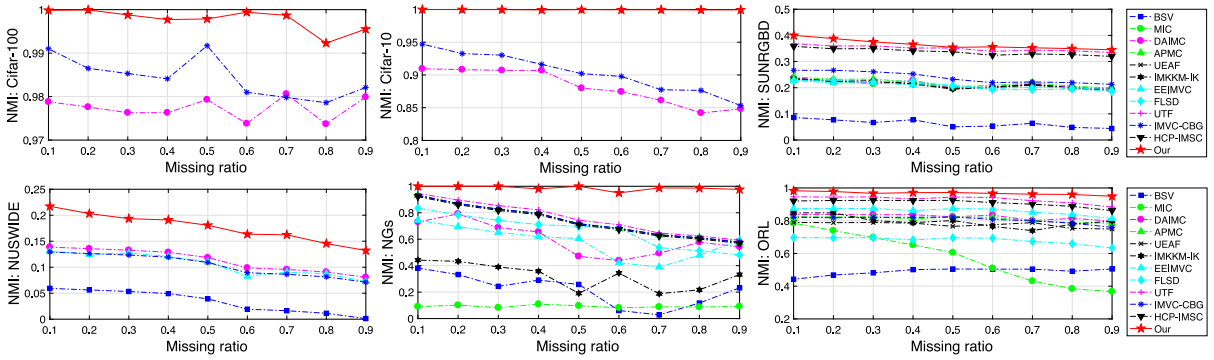


Fig. 5. The averaged NMI variation with missing ratios from 0.1 to 0.9 on six benchmark datasets are reported.

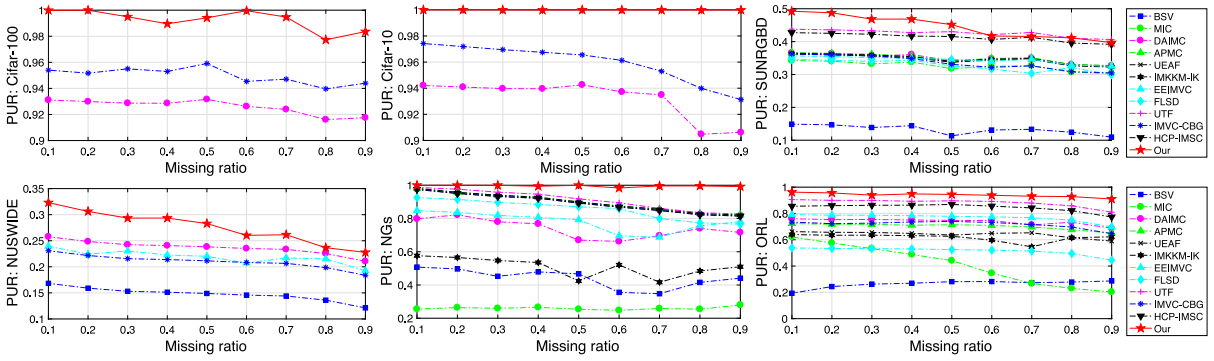


Fig. 6. The averaged PUR variation with missing ratios from 0.1 to 0.9 on six benchmark datasets are reported.

4.5. Parameter validity analysis

Our Algorithm 1 is involved in three parameters required to be set suitably, *i.e.*, parameters β, γ, and anchor number m. From Fig. 9, we perform grid search on the large-scale Cifar10 dataset, we first vary parameter β and γ in $2^{[-5:1:5]}$ with the fixed $m = 2c$. Then, we tune m in $[2c, 3c, 5c, 7c, 10c]$ with the fixed $\beta = 2^{-4}$ and $\gamma = 2^{3}$. Fig. 9(a) indicates that our CGMAA enjoys a satisfying clustering performance in a large scope *w.r.t* β and γ. Thus, Algorithm 1 is insensitive to β and γ. And Fig. 9(b) shows the excellent and stable clustering performance

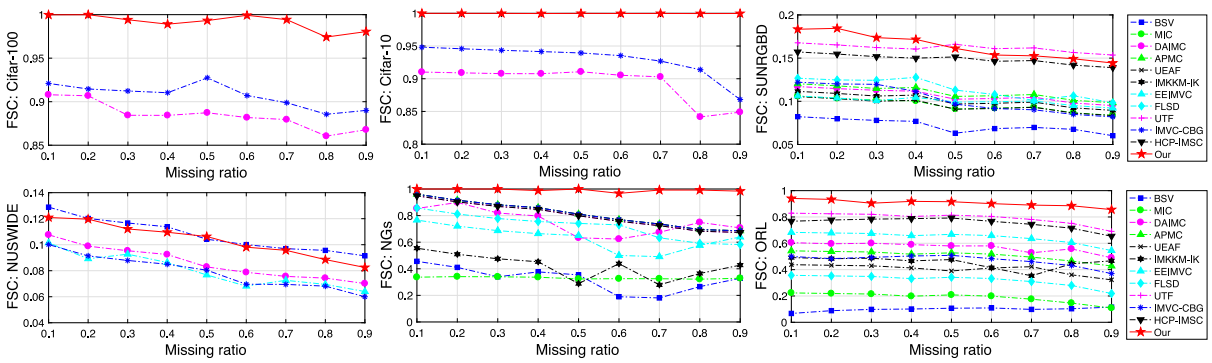
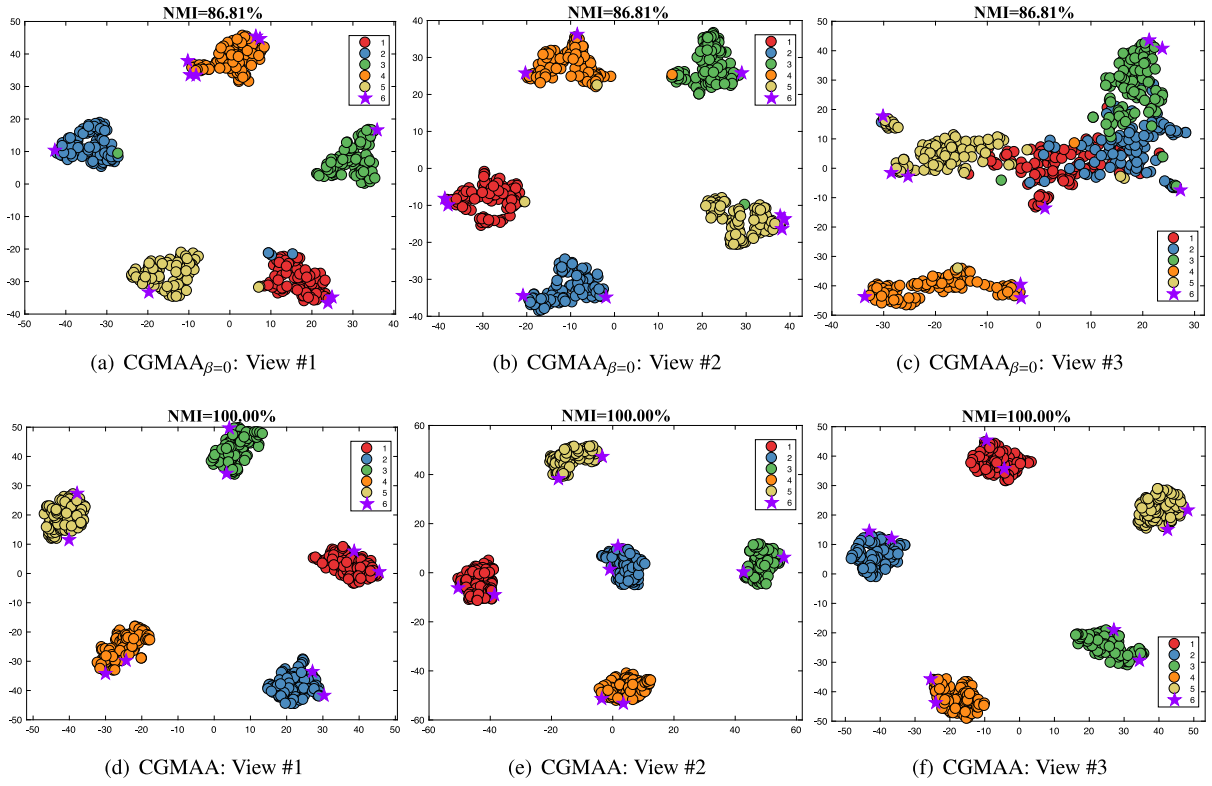


Fig. 7. The averaged FSC variation with missing ratios from 0.1 to 0.9 on six benchmark datasets are reported.

Table 5
Clustering comparison of CGMAA and CGMAA$_{\beta=0}$ on the NGs dataset *w.r.t* different missing ratios.

Method	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
ACC									
CGMAA$_{\beta=0}$	89.79 ± 0.01	90.68 ± 0.00	94.60 ± 0.01	91.80 ± 0.01	94.40 ± 0.00	95.27 ± 0.00	94.20 ± 0.00	95.40 ± 0.00	95.00 ± 0.00
CGMAA	**99.20 ± 0.00**	**99.60 ± 0.00**	**99.60 ± 0.00**	**98.36 ± 0.01**	**100.00 ± 0.00**	**99.40 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
NMI									
CGMAA$_{\beta=0}$	74.05 ± 0.01	75.69 ± 0.00	84.42 ± 0.00	80.78 ± 0.01	85.13 ± 0.00	86.42 ± 0.00	85.18 ± 0.00	86.81 ± 0.00	85.81 ± 0.00
CGMAA	**97.62 ± 0.00**	**98.61 ± 0.00**	**98.77 ± 0.00**	**94.99 ± 0.01**	**100.00 ± 0.00**	**98.08 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**



(a) CGMAA$_{\beta=0}$: View #1 (b) CGMAA$_{\beta=0}$: View #2 (c) CGMAA$_{\beta=0}$: View #3

(d) CGMAA: View #1 (e) CGMAA: View #2 (f) CGMAA: View #3

Fig. 8. The anchor distribution of the proposed CGMAA$_{\beta=0}$ and CGMAA is visualized with *t*-SNE [42] on NGs datasets (ψ=0.2).

when m in range $[2c, 3c, 5c, 7c, 10c]$. This demonstrates the validity of cross-view graph matching guided anchor learning.

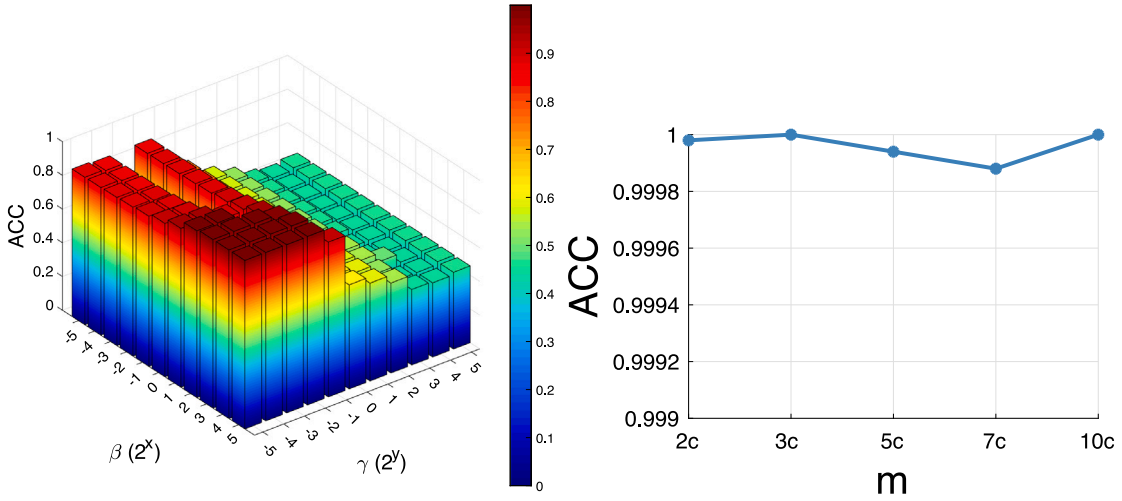
The running time of IMVC methods on the incomplete benchmark datasets is reported in Table 7. Table 7 and Fig. 10 demonstrate that our CGMAA can achieve very competitive computational efficiency on all evaluated datasets and competitors. Overall, CGMAA achieves a lower time cost and space complexity, as well as a very competitive clustering performance.

4.6. Conclusion

In this paper, we observe a cross-view anchor misalignment (CAM) problem in the process of anchor learning. And then, a novel CGMAA framework is proposed to handle the observed CAM problem for complete/incomplete multi-view bipartite graph clustering. Whether the data is missing or not, CGMAA can efficiently and flexibly address the CAM problem. Besides, CGMAA has great potential for large-scale

Table 6
The performance comparison *w.r.t* FMVACC and our CGMAA. Best results are marked in bold.

Samples	2000	2500	10335	60000	400	30000
Method	UCI-Digit	BDGP	SUNRGBD	MNIST	ORL	NUSWIDE
ACC						
FMVACC	83.59 ± 7.16	59.51 ± 4.00	21.95 ± 0.73	94.39 ± 6.64	74.86 ± 2.36	15.35 ± 0.68
CGMAA	**99.40 ± 0.00**	**95.08 ± 0.00**	**25.06 ± 0.01**	**97.62 ± 0.01**	**97.58 ± 0.03**	**25.27 ± 0.01**
NMI						
FMVACC	81.13 ± 3.79	35.55 ± 4.53	19.22 ± 0.77	95.11 ± 2.52	86.26 ± 2.08	21.42 ± 0.91
CGMAA	**98.48 ± 0.00**	**88.58 ± 0.00**	**24.35 ± 0.00**	**96.33 ± 0.01**	**98.81 ± 0.02**	**40.33 ± 0.01**



(a) Cifar10 *w.r.t* α and β (b) Cifar10 *w.r.t* m

Fig. 9. ACC *w.r.t* α, β, and m on the Cifar10 dataset (ψ=0.2).

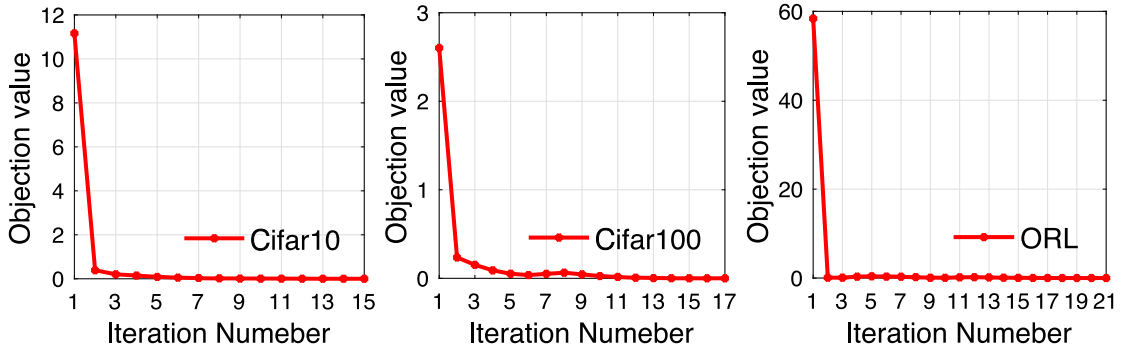


Fig. 10. Objection function value over the Cifar10, Cifar100, and ORL datasets (ψ=0.2).

Table 7
Running time on the different incomplete benchmarks. And "-" means out of the storage memory or out of the CPU memory.

Method	ORL	NGs	Caltech-20	SUNRGBD	NUSWIDE	Cifar10	Cifar100
BSV	0.97	0.29	2.32	2361.05	9624.38	–	–
MIC	418.37	143.94	2847.45	6107.6	–	–	–
DAIMC	1205.6	42.61	69.88	184.06	1667.67	29948.48	26349.92
APMC	0.58	0.61	–	156.37	–	–	–
UEAF	14.3	2.36	26.63	89.61	–	–	–
IMKKM-IK	0.58	1.47	126.15	309.29	–	–	–
EE-R-IMVC	0.73	0.57	3.96	5.68	1872.44	–	–
FLSD	87.44	3.41	54.04	130.79	–	–	–
UTF	10.72	6.32	60.38	209.34	–	–	–
IMVC-CBG	1.63	1.42	5.39	8.61	22.54	69.18	155.62
HCP-IMSC	8.37	6.32	157.38	358.34	–	–	–
Our	2.45	1.24	5.99	16.19	39.65	57.25	146.86

multi-view clustering tasks. Comprehensive experiments are performed to show the efficiency and superiority of cross-view anchor alignment and the proposed CGMAA framework. CAM is a pioneering problem for view-specific anchor learning, and the study of CAM can further benefit the multi-view bipartite graph clustering and research community.

CRediT authorship contribution statement

Xingfeng Li: Conceptualization, Methodology, Software, Visualization, Writing – original draft. **Yinghui Sun:** Data curation, Writing – original draft. **Quansen Sun:** Review, Supervision. **Zhenwen Ren:** Validation, Review & editing. **Yuan Sun:** Review, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61673220 and 62106209), the Base Strengthening Program of National Defense Science and Technology (Grant no. 2022-JCJQ-JJ-0292), the Grant from Guangxi Key Laboratory of Machine Vision and Intelligent Control (Grant no. 2022B07), the Sichuan Science and Technology Program (Grant no. 2023NSFSC0506), the Open Research Fund of Anhui Province Key Laboratory of Machine Vision Inspection (Grant no. KLMVI-2023-HIT-08), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant no. KYCX23_0487).

References

[1] X. Li, Q. Sun, Z. Ren, Y. Sun, Dynamic incomplete multi-view imputing and clustering, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 3412–3420.
[2] J. You, Z. Ren, Q. Sun, Y. Sun, X. Li, Approximate shifted Laplacian reconstruction for multiple kernel clustering, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 2862–2870.
[3] X. Li, Z. Ren, Q. Sun, Z. Xu, Auto-weighted tensor schatten p-norm for robust multi-view graph clustering, Pattern Recognit. 134 (2023) 109083.
[4] J. You, Z. Ren, F.R. Yu, X. You, One-stage shifted Laplacian refining for multiple kernel clustering, IEEE Trans. Neural Netw. Learn. Syst. (2023) 1–13.
[5] Z. Fu, Y. Zhao, D. Chang, Y. Wang, J. Wen, Latent low-rank representation with weighted distance penalty for clustering, IEEE Trans. Cybern. (2022) 1–13.
[6] X. Li, Y. Sun, Q. Sun, Z. Ren, Consensus cluster center guided latent multi-kernel clustering, IEEE Trans. Circuits Syst. Video Technol. (2022).
[7] X. Liu, L. Liu, Q. Liao, S. Wang, Y. Zhang, W. Tu, C. Tang, J. Liu, E. Zhu, One pass late fusion multi-view clustering, in: International Conference on Machine Learning, PMLR, 2021, pp. 6850–6859.
[8] Y. Chen, S. Wang, C. Peng, Z. Hua, Y. Zhou, Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering, IEEE Trans. Image Process. 30 (2021) 4022–4035.
[9] Y. Chen, X. Xiao, C. Peng, G. Lu, Y. Zhou, Low-rank tensor graph learning for multi-view subspace clustering, IEEE Trans. Circuits Syst. Video Technol. 32 (1) (2021) 92–104.
[10] M.-S. Chen, L. Huang, C.-D. Wang, D. Huang, J.-H. Lai, Relaxed multi-view clustering in latent embedding space, Inf. Fusion 68 (2021) 8–21.
[11] X. Li, Y. Sun, Q. Sun, Z. Ren, Enforced block diagonal graph learning for multikernel clustering, IEEE Trans. Comput. Soc. Syst. (2023) 1–13.
[12] X. Cai, D. Huang, G.-Y. Zhang, C.-D. Wang, Seeking commonness and inconsistencies: A jointly smoothed approach to multi-view subspace clustering, Inf. Fusion 91 (2023) 364–375.
[13] J. Wen, Z. Zhang, L. Fei, B. Zhang, Y. Xu, Z. Zhang, J. Li, A survey on incomplete multi-view clustering, IEEE Trans. Syst. Man Cybern.: Syst. (2022).
[14] M.-S. Chen, C.-D. Wang, D. Huang, J.-H. Lai, P.S. Yu, Efficient orthogonal multi-view subspace clustering, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 127–135.
[15] W. Xia, Q. Gao, Q. Wang, X. Gao, C. Ding, D. Tao, Tensorized bipartite graph learning for multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. (2022) 1–16.
[16] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, Z. Xu, Large-scale multi-view subspace clustering in linear time, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 4412–4419.
[17] X. Li, H. Zhang, R. Wang, F. Nie, Multiview clustering: A scalable and parameter-free bipartite graph fusion method, IEEE Trans. Pattern Anal. Mach. Intell. 44 (1) (2020) 330–344.
[18] S. Wang, X. Liu, X. Zhu, P. Zhang, Y. Zhang, F. Gao, E. Zhu, Fast parameter-free multi-view subspace clustering with consensus anchor guidance, IEEE Trans. Image Process. 31 (2021) 556–568.
[19] W. Shao, L. He, P.S. Yu, Multiple incomplete views clustering via weighted nonnegative matrix factorization with L2,1 regularization, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2015, pp. 318–334.
[20] M. Hu, S. Chen, Doubly aligned incomplete multi-view clustering, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-19, 2019, pp. 2262–2268.
[21] J. Wen, H. Sun, L. Fei, J. Li, Z. Zhang, B. Zhang, Consensus guided incomplete multi-view spectral clustering, Neural Netw. 133 (2021) 207–219.
[22] J. Wen, Z. Zhang, Z. Zhang, L. Fei, M. Wang, Generalized incomplete multiview clustering with flexible locality structure diffusion, IEEE Trans. Cybern. 51 (1) (2020) 101–114.
[23] J. Wen, Z. Zhang, Y. Xu, B. Zhang, L. Fei, H. Liu, Unified embedding alignment with missing views inferring for incomplete multi-view clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 5393–5400.
[24] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, W. Gao, Late fusion incomplete multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 41 (10) (2018) 2410–2423.
[25] X. Zhu, X. Liu, M. Li, E. Zhu, L. Liu, Z. Cai, J. Yin, W. Gao, Localized incomplete multiple kernel k-means., in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-18, 2018, pp. 3271–3277.
[26] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, W. Gao, Multiple kernel k-means with incomplete kernels, IEEE Trans. Pattern Anal. Mach. Intell. 42 (5) (2019) 1191–1204.
[27] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, E. Zhu, Efficient and effective regularized incomplete multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 43 (8) (2020) 2634–2646.
[28] Y. Zhang, X. Liu, S. Wang, J. Liu, S. Dai, E. Zhu, One-stage incomplete multi-view clustering via late fusion, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 2717–2725.
[29] W. He, Z. Zhang, Y. Chen, J. Wen, Structured anchor-inferred graph learning for universal incomplete multi-view clustering, World Wide Web (2022) 1–25.
[30] J. Wen, Y. Xu, H. Liu, Incomplete multiview spectral clustering with adaptive graph learning, IEEE Trans. Syst. Man Cybern. 50 (4) (2020) 1418–1429.
[31] J. Wen, Z. Zhang, Z. Zhang, L. Zhu, L. Fei, B. Zhang, Y. Xu, Unified tensor framework for incomplete multi-view clustering and missing-view inferring, in: Proc. of the 35th AAAI Conference on Artificial Intelligence, Online: AAAI Press, 2021, pp. 10273–10281.
[32] J. Liu, X. Liu, Y. Zhang, P. Zhang, W. Tu, S. Wang, S. Zhou, W. Liang, S. Wang, Y. Yang, Self-representation subspace clustering for incomplete multi-view data, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 2726–2734.
[33] Z. Li, C. Tang, X. Zheng, X. Liu, W. Zhang, E. Zhu, High-order correlation preserved incomplete multi-view subspace clustering, IEEE Trans. Image Process. 31 (2022) 2067–2080.
[34] S. Wang, X. Liu, S. Liu, J. Jin, W. Tu, X. Zhu, E. Zhu, Align then fusion: Generalized large-scale multi-view clustering with anchor matching correspondences, 2022, arXiv preprint arXiv:2205.15075.
[35] Z. Ren, Q. Sun, D. Wei, Multiple kernel clustering with kernel k-means coupled graph tensor learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 9411–9418.
[36] Y. Sun, D. Peng, H. Huang, Z. Ren, Feature and semantic views consensus hashing for image set classification, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 2097–2105.
[37] Z. Ren, S.X. Yang, Q. Sun, T. Wang, Consensus affinity graph learning for multiple kernel clustering, IEEE Trans. Cybern. 51 (6) (2020) 3273–3284.
[38] Y. Sun, Z. Ren, P. Hu, D. Peng, X. Wang, Hierarchical consensus hashing for cross-modal retrieval, IEEE Trans. Multimed. (2023) 1–14.
[39] F. Nie, J. Li, X. Li, et al., Self-weighted multiview clustering with multiple graphs., in: International Joint Conference on Articial Intelligence, 2017, pp. 2564–2570.
[40] J. Guo, J. Ye, Anchors bring ease: An embarrassingly simple approach to partial multi-view clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 118–125.
[41] S. Wang, X. Liu, L. Liu, W. Tu, X. Zhu, J. Liu, S. Zhou, E. Zhu, Highly-efficient incomplete large-scale multi-view clustering with consensus bipartite graph, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9776–9785.
[42] Y. Sun, X. Wang, D. Peng, Z. Ren, X. Shen, Hierarchical hashing learning for image set classification, IEEE Trans. Image Process. 32 (2023) 1732–1744.