Cross-View Approximation on Grassmann Manifold for Multiview Clustering

Yidan Ma, Xinjie Shen, Danyang Wu, Jianfu Cao, and Feiping Nie, *Senior Member, IEEE*

***Abstract*— In existing multiview clustering research, the comprehensive learning from multiview graph and feature spaces simultaneously remains insufficient when achieving a consistent clustering structure. In addition, a postprocessing step is often required. In light of these considerations, a cross-view approximation on Grassman manifold (CAGM) model is proposed to address inconsistencies within multiview adjacency matrices, feature matrices, and cross-view combinations from the two sources. The model uses a ratio-formed objective function, enabling parameter-free bidirectional fusion. Furthermore, the CAGM model incorporates a paired encoding mechanism to generate low-dimensional and orthogonal cross-view embeddings. Through the approximation of two measurable subspaces on the Grassmann manifold, the direct acquisition of the indicator matrix is realized. Furthermore, an effective optimization algorithm corresponding to the CAGM model is derived. Comprehensive experiments on four real-world datasets are conducted to substantiate the effectiveness of our proposed method.**

***Index Terms*— Adaptively weighted learning, bidirectional fusion, Grassmann manifold, multiview clustering.**

I. Introduction

Multiview clustering research seeks to leverage multiview data to establish a coherent clustering structure within an unsupervised framework [1], [2], [3], [4], [5], [6]. Presently, the existing methodologies can be broadly classified into two categories: one that focuses on generating consistent embeddings from multiple graph spaces [7], [8], [9], [10], [11], [12] and the other from multiple feature spaces [9], [13], [14], [15], [16], [17]. However, these unidirectional learning mechanisms have not sufficiently delved into the intricate interplay between graph structures and features.

Many researchers have developed various fusion strategies with the aim of reducing information loss and improving clustering performance. For example, some advancements focus on integrating latent representations from multiview data and clustering tasks to formulate a one-step strategy [18], [19]. In addition, clustering partition matrices and kernel matrices can be jointly learned to complement each other [20], [21]. Others unify graph learning, graph fusion, and spectral clustering [22], or use random view groups and an early–late fusion strategy to capture the relationship from multiple views [23]. Some algorithms begin to mitigate the gap between multiview feature spaces and graph spaces [24], [25]. While prior contributions have been established using attributed or bipartite graphs to leverage both the node attributes and graph structures, a notable deficiency exists in achieving an in-depth integration of graph structures and features across different views. Furthermore, the computational challenges stemming from the integration of cross-view combinations and high-dimensional features have not been adequately addressed.

In this study, we propose a model called cross-view approximation on the Grassmann manifold (CAGM) to improve effective interactions and information acquisition from both multiview graph structures and features while not introducing additional computational complexity. CAGM uses a bidirectional fusion method to obtain the indicator matrix directly from multiple adjacency and feature matrices. CAGM learns a weighted structural embedding from multiple adjacency matrices initially and generates cross-view low-dimensional embeddings from multiple adjacency and feature matrices using a paired encoding mechanism. Finally, the integration of aforementioned two steps is achieved by a ratio-based objective function, which introduces subspace approximation on the Grassmann manifold. CAGM facilitates interactions among multiview graph structures and features, preserving the integrity of the initial graph structure and feature information without requiring hyperparameter tuning or additional postprocessing, such as the K-means clustering algorithm. The robustness of the method is validated through experiments on diverse real-world datasets. The key contributions of this study are outlined as follows.

1) The CAGM model directly learns clustering results and improves unified feature extraction capabilities in both multiview graph space and feature space through a bidirectional fusion strategy, while preserving the integrity of the original features and graphs.
2) A comprehensive adaptive weighted learning scheme is devised to credit and ensure the complementarity of both multiview graph structures and features in the same view and cross-view combinations.
3) An alternating iterative algorithm is formulated to address the optimization challenge within the CAGM model, and experiments are conducted to affirm its efficiency on four real-world datasets.

Notations: The sample size, number of classes, and feature dimensionality for the vth view are denoted as n, c, and d_v, respectively. We analyze a multiview dataset comprising m_1 adjacency matrices $\{\mathbf{A}_{v_1}\}_{v_1=1}^{m_1}, \mathbf{A}_{v_1} \in \mathbb{R}^{n\times n}$ and m_2 feature matrices $\{\mathbf{X}_{v_2}\}_{v_2=1}^{m_2}, \mathbf{X}_{v_2} \in \mathbb{R}^{n\times d_{v_2}}$. We assume no discernible correlation between each view represented by $\mathbf{A}_{v_1}$ and $\mathbf{X}_{v_2}$. $\mathbf{Ind}(n, c)$ represents n by c indicator matrix, $\mathbb{R}$ refers to the real space, $\mathbf{I}$ stands for the identity matrix, and $\mathbf{X}^T$ and $\mathbf{Tr}(\mathbf{X})$ denote the transpose and trace of matrix $\mathbf{X}$, respectively. $\|\mathbf{X}\|_F = (\mathbf{Tr}(\mathbf{X}^T\mathbf{X}))^{1/2}$ denotes the Frobenius-norm of matrix $\mathbf{X}$, and $\|\boldsymbol{x}\|_2 = (\sum_i x_i^2)^{1/2}$ denotes the ℓ_2-norm of vector $\boldsymbol{x}$. We also use the notation $\|\boldsymbol{x}\|_{-1}$ to signify that it is equivalent to $\sum_i x_i^{-1} = 1$. $\boldsymbol{x} \succeq \mathbf{0}$ indicates that all the elements in vector $\boldsymbol{x}$ are greater than or equal to zero.

Manuscript received 2 November 2023; revised 19 March 2024; accepted 8 April 2024. This work was supported in part by the Key Research and Development Program of Xinjiang Uygur Autonomous Region under Grant 2022B01008, in part by the Natural Science Foundation of Henan Province under Grant 242300421417, and in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2024JC-YBQN-0700. *(Yidan Ma and Xinjie Shen are co-first authors.) (Corresponding author: Danyang Wu.)*

Yidan Ma and Jianfu Cao are with the State Key Laboratory for Manufacturing Systems Engineering and the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: mmmyd09@gmail.com; cjf@xjtu.edu.cn).

Xinjie Shen is with the School of Future Technology, South China University of Technology, Guangzhou, Guangdong 510640, China (e-mail: frinkleko@gmail.com).

Danyang Wu is with the College of Information Engineering and Shaanxi Engineering Research Center for Intelligent Perception and Analysis of Agricultural Information, Northwest A&F University, Xianyang, Shaanxi 710049, China (e-mail: danyangwu.cs@gmail.com).

Feiping Nie is with the School of Computer Science and the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: feipingnie@gmail.com).

Digital Object Identifier 10.1109/TNNLS.2024.3388192

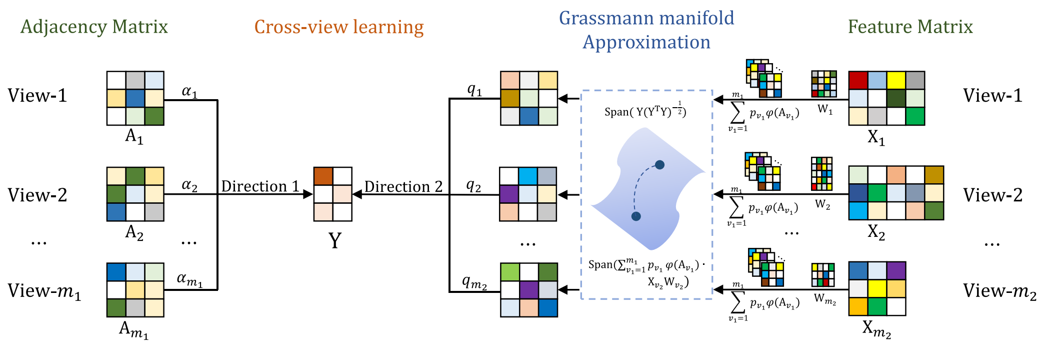


Fig. 1. Workflow of the proposed CAGM model.

II. Methodology

This section introduces the CAGM model, which obtains bidirectional interactions between multiview graph structures and features. Fig. 1 illustrates the workflow of the CAGM model. The specifics of acquiring the clustering indicator matrix are outlined as follows.

A. From the Side of Multiview Graph Structures

The initial objective is to optimize a weighted structural embedding $\mathbf{H}$ as $\mathbf{H} \in \mathbb{R}^{n\times c}$ approximating adjacency matrices $\{\mathbf{A}_{v_1}\}_{v_1=1}^{m_1}$ as close as possible. To enhance the separability of the acquired embedding, $\mathbf{H}$ is replaced with a combinatorial formulation of the indicator matrix $\mathbf{Y}$ as $\mathbf{H} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1/2}$, where $\mathbf{Y} \in \mathbf{Ind}(n, c)$ [26]. We also introduce an adjusting weight vector $\boldsymbol{\alpha}$ with the constraint $\|\boldsymbol{\alpha}\|_2 = 1, \boldsymbol{\alpha} \succeq \mathbf{0}$ to identify the importance of each $\mathbf{A}_{v_1}$. The problem can be formulated as follows:

$$\max_{\mathbf{Y},\boldsymbol{\alpha}} \sum_{v_1=1}^{m_1} \alpha_{v_1} \mathbf{Tr}\Big((\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\Big)$$
$$\text{s.t. } \mathbf{Y} \in \mathbf{Ind}(n, c), \quad \|\boldsymbol{\alpha}\|_2 = 1, \ \boldsymbol{\alpha} \succeq \mathbf{0} \tag{1}$$

where α_{v_1} is the v_1th element of $\boldsymbol{\alpha}$.

B. From the Side of Multiview Graph Features

In problem (1), $\mathbf{Y}$ is obtained unidirectionally from multiview graph structures. In this part, we develop a suitable encoding scheme to achieve interaction between multiview graph structures and features while maintaining the integrity of the original features and graphs.

Arbitrary graph filters $\varphi(\cdot)$ can be applied initially to multiple adjacency matrices $\{\mathbf{A}_{v_1}\}_{v_1=1}^{m_1}$ obtaining the structural representations $\{\varphi(\mathbf{A}_{v_1})\}_{v_1=1}^{m_1}$. We choose $\varphi(\mathbf{A}_{v_1}) = ((\mathbf{D}_{v_1}^{-1/2}\mathbf{A}_{v_1}\mathbf{D}_{v_1}^{-1/2} + \mathbf{I})/2)^g$, $g = 1$ in the experimental setup, where g is the filter order and $\{\mathbf{D}_{v_1}\}_{v_1=1}^{m_1}$ are the degree matrices of $\{\mathbf{A}_{v_1}\}_{v_1=1}^{m_1}$. Furthermore, we introduce $m_1 \times m_2$ linear projection matrices $\mathbf{W}_{v_1,v_2}$, $\mathbf{W}_{v_1,v_2} \in \mathbb{R}^{d_{v_2}\times c}$ to integrate each $\{\varphi(\mathbf{A}_{v_1})\}_{v_1=1}^{m_1}$ and $\{\mathbf{X}_{v_2}\}_{v_2=1}^{m_2}$, yielding low-dimensional cross-view representations $\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_1,v_2}$. Considering the perspective of multiview features, we introduce an adjusting weight $\boldsymbol{p}$ to obtain m_2 target orthogonal embeddings $\sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2}$, where p_{v_1} is the v_1th element of $\boldsymbol{p}$, $\|\boldsymbol{p}\|_2 = 1$. Each $\mathbf{W}_{v_1,v_2}$ is simplified as $\mathbf{W}_{v_2}$ to illustrate the projection for $\sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}$. We simplify each $\sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2}$ as $\hat{\mathbf{H}}_{v_2}$, which satisfies the orthogonality constraint $\hat{\mathbf{H}}_{v_2}^T\hat{\mathbf{H}}_{v_2} = \mathbf{I}$.

To achieve the feasible similarity between the desired indicator matrix $\mathbf{Y}$ and the paired encodings $\hat{\mathbf{H}}_{v_2}$, we approach this task as an approximation learning problem involving two subspaces on the Grassmann manifold. Since $\hat{\mathbf{H}}_{v_2}$ and $\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1/2}$ adhere to the orthogonal constraint $\hat{\mathbf{H}}_{v_2}^T\hat{\mathbf{H}}_{v_2} = \mathbf{I}$ and $\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T = \mathbf{I}$, respectively, we can describe the distance between the two subspaces $\mathbf{span}(\hat{\mathbf{H}}_{v_2})$ and $\mathbf{span}(\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1/2})$ by the squared projection metric [27]. In addition, we introduce another adjusting weight $\boldsymbol{q}$ with the constraint $\|\boldsymbol{q}\|_{-1} = 1, \boldsymbol{q} \succeq \mathbf{0}$ to represent the significance of embeddings, as shown in the following equation:

$$\min_{\mathbf{Y},\boldsymbol{p},\boldsymbol{q},\mathbf{W}} \sum_{v_2=1}^{m_2} q_{v_2}\big\|\hat{\mathbf{H}}_{v_2}\hat{\mathbf{H}}_{v_2}^T - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\big\|_F^2$$
$$\text{s.t. } \mathbf{Y} \in \mathbf{Ind}(n, c), \quad \|\boldsymbol{q}\|_{-1} = 1, \ \boldsymbol{q} \succeq \mathbf{0}, \ \|\boldsymbol{p}\|_2 = 1, \ \hat{\mathbf{H}}_{v_2}^T\hat{\mathbf{H}}_{v_2} = \mathbf{I}$$
$$\hat{\mathbf{H}}_{v_2} = \sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2}, \quad v_2 \in [1, m_2] \tag{2}$$

where q_{v_2} corresponds to the v_2th element of $\boldsymbol{q}$. Consequently, a cross-view uniform embedding can be effectively achieved.

C. Bidirectional Fusion

This section aims to establish a bidirectional learning between problem (1) and problem (2). Compared with introducing hyperparameters to link the two learning tasks, adopting a ratio-based approach is more suitable for real applications, as it helps avoid manual tuning [28]. Mathematically, the problem can be reformulated as follows:

$$\max_{\mathbf{Y},\boldsymbol{\alpha},\boldsymbol{p},\boldsymbol{q},\mathbf{W}} \frac{\sum_{v_1=1}^{m_1} \alpha_{v_1}\mathbf{Tr}\Big((\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}Y(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\Big)}{\sum_{v_2=1}^{m_2} q_{v_2}\big\|\hat{\mathbf{H}}_{v_2}\hat{\mathbf{H}}_{v_2}^T - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\big\|_F^2}$$
$$\text{s.t. } \mathbf{Y} \in \mathbf{Ind}(n, c), \quad \|\boldsymbol{\alpha}\|_2 = 1, \ \boldsymbol{\alpha} \succeq \mathbf{0}, \ \|\boldsymbol{p}\|_2 = 1, \ \|\boldsymbol{q}\|_{-1} = 1$$
$$\boldsymbol{q} \succeq \mathbf{0}, \quad \hat{\mathbf{H}}_{v_2} = \sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2}, \ \hat{\mathbf{H}}_{v_2}^T\hat{\mathbf{H}}_{v_2} = \mathbf{I}$$
$$v_2 \in [1, m_2]. \tag{3}$$

Problem (3) aims to learn the uniform clustering structure by fully exploring the information from the combination of cross-view graph structures $\{\mathbf{A}_{v_1}\}_{v_1=1}^{m_1}$ and features $\{\mathbf{X}_{v_2}\}_{v_2=1}^{m_1}$.

III. Optimization

A. Related Framework

Within this segment, the formulation of an iterative algorithm is introduced to tackle problem (3). Initially, the procedure delineated as Algorithm 1 is applied to resolve the following general maximization ratio problem [29]:

$$\max_{\boldsymbol{x}\in\Delta} \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \tag{4}$$

where Δ is an arbitrary constraint on $\boldsymbol{x}$ and $g(\boldsymbol{x}) > 0$.

Algorithm 1 enhances the value of the objective function for problem (4) through a series of iterations until it converges. Moreover, the global solution of problem (4) aligns with the root of a subsequent function

$$\arg\max_{\boldsymbol{x}\in\Delta} f(\boldsymbol{x}) - \lambda_i g(\boldsymbol{x}). \tag{5}$$

Considering the denominator in problem (3) $\sum_{v_2=1}^{m_2} q_{v_2}\|\hat{\mathbf{H}}_{v_2}\hat{\mathbf{H}}_{v_2}^T - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\|_F^2$ is greater than 0, we can reformulate problem (3)

Algorithm 1 Algorithm to Solve Problem (4)

```
Initialization: $\boldsymbol{x} \in \Delta$, iteration step $i = 1$
while not converge do
    Calculate $\lambda_i = \frac{f(x_i)}{g(x_i)}$
    Calculate $x_{i+1} = \arg\max_{\boldsymbol{x}\in\Delta} f(\boldsymbol{x}) - \lambda_i g(\boldsymbol{x})$
    $i = i + 1$
endwhile
```

as follows:

$$\max_{\mathbf{Y},\boldsymbol{\alpha},\boldsymbol{p},\boldsymbol{q},\mathbf{W}} \sum_{v_1=1}^{m_1} \alpha_{v_1}\mathbf{Tr}\Big((\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\Big)$$
$$-\lambda\sum_{v_2=1}^{m_2} q_{v_2}\left\|\hat{\mathbf{H}}_{v_2}\hat{\mathbf{H}}_{v_2}^T - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\right\|_F^2$$
$$\text{s.t. } \mathbf{Y}\in\mathbf{Ind}(n,c),\quad \|\boldsymbol{\alpha}\|_2 = 1,\ \ \boldsymbol{\alpha}\succeq\mathbf{0},\ \|\boldsymbol{p}\|_2 = 1,\ \ \|\boldsymbol{q}\|_{-1} = 1$$
$$\boldsymbol{q}\succeq\mathbf{0},\quad \hat{\mathbf{H}}_{v_2} = \sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2},\ \ \hat{\mathbf{H}}_{v_2}^T\hat{\mathbf{H}}_{v_2} = \mathbf{I}$$
$$v_2\in[1,m_2]. \tag{6}$$

Following this, problem (6) can be solved via the following iterative procedure. A comprehensive adaptive weighted learning scheme that incorporates $\boldsymbol{\alpha}$, $\boldsymbol{q}$, and $\boldsymbol{p}$ is developed to assign credit to multiview graph structures and features.

B. Derivation

In each iteration, λ is computed based on the current $\mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{p}, \boldsymbol{q}$ and $\mathbf{W}$, constituting the outer-loop optimization. In the subsequent inner-loop optimization, $\mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{p}, \boldsymbol{q}$ and $\mathbf{W}$ are alternately optimized alongside the current λ.

Update $\mathbf{Y}$ and fix $\boldsymbol{\alpha}, \boldsymbol{p}, \boldsymbol{q}, \mathbf{W}$. When $\mathbf{R}_1 = \sum_{v_1=1}^{m_1}\alpha_{v_1}\mathbf{A}_{v_1}$ and $\mathbf{R}_2 = 2\sum_{v_2=1}^{m_2} q_{v_2}(\sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2})(\sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2})^T$ are denoted, problem (6) can be rewritten as the following trace maximum problem:

$$\max_{\mathbf{Y}\in\mathbf{Ind}} \mathbf{Tr}\Big(\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\mathbf{Y}^T(\mathbf{R}_1+\lambda\mathbf{R}_2)\mathbf{Y}\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\Big). \tag{7}$$

For $\mathbf{Y}\in\mathbf{Ind}(n,c)$ signifying $\mathbf{Y}$ is row-independent, we can use the fast coordinate descend algorithm to handle problem (7) [26].

Update $\boldsymbol{\alpha}$ and fix $\mathbf{Y}, \boldsymbol{p}, \boldsymbol{q}, \mathbf{W}$. Problem (6) with respect to $\boldsymbol{\alpha}$ is transformed into the following problem:

$$\max_{\boldsymbol{\alpha}} \sum_{v_1=1}^{m_1} \alpha_{v_1}\mathbf{Tr}\Big(\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\Big)$$
$$\text{s.t. } \|\boldsymbol{\alpha}\|_2 = 1,\quad \boldsymbol{\alpha}\succeq\mathbf{0}. \tag{8}$$

For convenience, we denote $\{\mathbf{h}_{v_1}\}_{v_1=1}^{m_1} = \{\alpha_{v_1}^2\}_{v_1=1}^{m_1}$ and rewrite problem (8) as follows:

$$\max_{\boldsymbol{h}} \sum_{v_1=1}^{m_1} \sqrt{h_{v_1}}\mathbf{Tr}\Big(\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\Big)$$
$$\text{s.t. } \sum_{v_1=1}^{m_1} h_{v_1} = 1,\quad \boldsymbol{h}\succeq\mathbf{0}. \tag{9}$$

Considering the term $\sum_{v_1=1}^{m_1} h_{v_1} = 1$, the following results using Cauchy–Schwarz inequality can be obtained [30]:

$$\sum_{v_1=1}^{m_1} \sqrt{h_{v_1}}\mathbf{Tr}\Big(\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\Big)$$
$$\overset{(a)}{\leq} \sqrt{\sum_{v_1=1}^{m_1} h_{v_1}}\sqrt{\sum_{v_1=1}^{m_1}\Big(\mathbf{Tr}\Big(\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\Big)\Big)^2}. \tag{10}$$

With constant r, the equality in (a) is satisfied when $h_{v_1} = r(\mathbf{Tr}((\mathbf{Y}^T\mathbf{Y})^{-1/2}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1/2}))^2$. Consequently, we can determine the specific value of r

$$r = \frac{1}{\sum_{v_1=1}^{m_1}\Big(\mathbf{Tr}\Big(\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\Big)\Big)^2}. \tag{11}$$

The value of r can be used to derive h_{v_1} and then use h_{v_1} for the calculation of α_{v_1} as shown below

$$\alpha_{v_1} = \frac{\mathbf{Tr}\Big(\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\Big)}{\sqrt{\sum_{v_1=1}^{m_1}\Big(\mathbf{Tr}\Big(\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}\big(\mathbf{Y}^T\mathbf{Y}\big)^{-\frac{1}{2}}\Big)\Big)^2}} \tag{12}$$

which corresponds to the solution for the v_1th view in problem (8).

Update $\boldsymbol{q}$ and fix $\mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{p}, \mathbf{W}$. Problem (6) concerning $\boldsymbol{q}$ can be expressed as follows:

$$\min_{\boldsymbol{q}} \sum_{v_2=1}^{m_2} q_{v_2}\left\|\hat{\mathbf{H}}_{v_2}\hat{\mathbf{H}}_{v_2}^T - \mathbf{Y}\big(\mathbf{Y}^T\mathbf{Y}\big)^{-1}\mathbf{Y}^T\right\|_F^2$$
$$\text{s.t. } \|\boldsymbol{q}\|_{-1} = 1, \boldsymbol{q}\succeq 0 \tag{13}$$

where $\hat{\mathbf{H}}_{v_2} = \sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2}$, $v_2\in[1,m_2]$. To simplify the notation, we denote $d_{v_2} = 1/q_{v_2}, r_{v_2} = \|\hat{\mathbf{H}}_{v_2}\hat{\mathbf{H}}_{v_2}^T - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\|_F^2$, where $v_2\in[1,m_2]$. Consequently, problem (13) can be reformulated as the following expression with respect to $\boldsymbol{d}$:

$$\min_{\sum_{v_2=1}^{m_2} d_{v_2}=1,\boldsymbol{d}\geq 0} \sum_{v_2=1}^{m_2}\frac{r_{v_2}}{d_{v_2}}. \tag{14}$$

Based on the constraints $\sum_{v_2=1}^{m_2} d_{v_2} = 1, \boldsymbol{d} \succeq \mathbf{0}$ and the Cauchy–Schwarz inequality, we can derive the lower bound for the objective in problem (14)

$$\sum_{v_2=1}^{m_2}\frac{r_{v_2}}{d_{v_2}} = \sum_{v_2=1}^{m_2}\frac{r_{v_2}}{d_{v_2}}\sum_{v_2=1}^{m_2} d_{v_2} \overset{(b)}{\geq} \left(\sum_{v_2=1}^{m_2}\sqrt{r_{v_2}}\right)^2 \tag{15}$$

where the equality in (b) holds when $d_{v_2} = e\sqrt{r_{v_2}}$ with constant t, $v_2\in[1,m_2]$. After obtaining e through the equality in (b), we can derive each q_{v_2} in order

$$q_{v_2} = \frac{\sum_{v_2=1}^{m_2}\sqrt{r_{v_2}}}{\sqrt{r_{v_2}}},\quad v_2\in[1,m_2] \tag{16}$$

where $\boldsymbol{q}$ is the solution of problem (13).

Update $\mathbf{W}$ and fix $\mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{p}, \boldsymbol{q}$. When $\mathbf{R}_1 = \mathbf{X}_{v_2}^T(\sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1}))^T\sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}$ and $\mathbf{R}_2 = \mathbf{X}_{v_2}^T(\sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1}))^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}$, where $v_1\in[1,m_1], v_2\in[1,m_2]$ are denoted, problem (6) with respect to each $\mathbf{W}_{v_2}$ becomes

$$\max_{\mathbf{W}_{v_2}^T\mathbf{R}_1\mathbf{W}_{v_2}=\mathbf{I}} \mathbf{Tr}\big(\mathbf{W}_{v_2}^T\mathbf{R}_2\mathbf{W}_{v_2}\big). \tag{17}$$

The solution to each $\mathbf{W}_{v_2}$ is formed by the eigenvectors corresponding to the first c-largest eigenvalues of $\mathbf{R}_1^{-1}\mathbf{R}_2$.

Update $\boldsymbol{p}$ and fix $\mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{q}, \mathbf{W}$. The reformulation of problem (6) with respect to $\boldsymbol{p}$ is as follows:

$$\max_{\|\boldsymbol{p}\|_2=1}\sum_{v_2=1}^{m_2} q_{v_2}\mathbf{Tr}\big(\hat{\mathbf{H}}_{v_2}\hat{\mathbf{H}}_{v_2}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\big) \tag{18}$$

where $\hat{\mathbf{H}}_{v_2} = \sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2}$, $v_2\in[1,m_2]$. Define $\mathbf{M}_{v_1} = \varphi(\mathbf{A}_{v_1})\sum_{v_2=1}^{m_2} q_{v_2}\mathbf{X}_{v_2}\mathbf{W}_{v_2}\mathbf{W}_{v_2}^T\mathbf{X}_{v_2}^T$ and $\mathbf{N}_{v_1} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\varphi(\mathbf{A}_{v_1})$, where $v_1\in[1,m_1]$, $v_2\in[1,m_2]$, reformulate problem (18) as

$$\max_{\|\boldsymbol{p}\|_2=1}\mathbf{Tr}\left(\left(\sum_{v_1=1}^{m_1} p_{v_1}\mathbf{M}_{v_1}\right)\left(\sum_{v_1=1}^{m_1} p_{v_1}\mathbf{N}_{v_1}\right)^T\right). \tag{19}$$

Introducing the column-based matrix vectorization operator $\mathbf{col}(\cdot)$ to acquire $\hat{\mathbf{M}} = [\mathbf{col}(\mathbf{M}_1),\ldots,\mathbf{col}(\mathbf{M}_{m_1})]$ and

Algorithm 2 Optimization Algorithm of CAGM

```
Input: $\mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{p}, \boldsymbol{q}, \mathbf{W}$
Output: $\mathbf{Y}$
while not converge do
    Calculate $\lambda$ via (3)
    while not converge do
        Update $\mathbf{Y}$ via solving Problem (7)
        Update $\boldsymbol{\alpha}$ via solving Problem (8)
        Update $\boldsymbol{q}$ via solving Problem (13)
        Update $\mathbf{W}$ via solving Problem (17)
        Update $\boldsymbol{p}$ via solving Problem (18)
    endwhile
endwhile
```

$\hat{\mathbf{N}} = [\mathbf{col}(\mathbf{N}_1), \ldots, \mathbf{col}(\mathbf{N}_{m_1})]$ as $\hat{\mathbf{M}}, \hat{\mathbf{N}} \in \mathbb{R}^{nm_1 \times m_1}$ [31]. Subsequently, the problem (19) can be rewritten as follows:

$$\max_{\|\boldsymbol{p}\|_2=1} \mathbf{Tr}\big(\boldsymbol{p}^T \hat{\mathbf{N}}^T \hat{\mathbf{M}} \boldsymbol{p}\big). \tag{20}$$

The solution for problem (20) can be formed as the eigenvectors corresponding to the m_1 largest eigenvalues of the matrix $\hat{\mathbf{N}}^T \hat{\mathbf{M}}$. The above derivation is summarized in Algorithm 2.

C. Theoretical Analysis

1) Computational Cost: Consider the values of $\{\mathbf{X}_{v_2}\}_{v_2=1}^{m_2}$ are dense and the filter parameter g in the graph filter $\varphi(\mathbf{A}_{v_1}) = ((\mathbf{D}_{v_1}^{-1/2}\mathbf{A}_{v_1}\mathbf{D}_{v_1}^{-1/2}+\mathbf{I})/2)^g$ has no effect on the parameter-free property since it is not in the proposed model.

Hence, we can determine the computational complexity associated with updating each variable. Update $\mathbf{Y}$: $O(n^2c + m_1n^2d_{v_2} + n^2d_{v_2})$; update $\boldsymbol{q}$: $O(n^3 + m_1n^2d_{v_2} + nd_{v_2}^2c + n)$; update $\boldsymbol{\alpha}$: $O(m_1n^2c)$; update $\boldsymbol{p}$: $O(n^2m_1^2c)$; and update each $\mathbf{W}_{v_2}$: $O(d_{v_2}^3 + n^2d_{v_2} + n)$. The computation of λ is only required during the initialization step and it can be recalculated when computing $\boldsymbol{\alpha}$ and $\boldsymbol{q}$ during the iterations. The whole computational cost: $O([(m_2 + cd_{v_2})^2n + (m_1^2 + m_1)cn^2 + (m_1 + m_2)d_{v_2}n^2 + n^3]T)$, where T is the number of iterations.

2) Convergence Analysis: we summarize the convergence guarantee into Theorem 1.

Theorem 1: The objective value of problem (3) will display a monotonically increasing trend until Algorithm 2 converges.

Proof: In the inner loop of Algorithm 2, $\mathbf{Y}_t$, $\boldsymbol{\alpha}_t$, $\boldsymbol{q}_t$, $\mathbf{W}_t$, and $\boldsymbol{p}_t$ in the tth iteration are updated to their respective counterparts, namely, $\mathbf{Y}_{t+1}$, $\boldsymbol{\alpha}_{t+1}$, $\boldsymbol{q}_{t+1}$, $\mathbf{W}_{t+1}$, and $\boldsymbol{p}_{t+1}$. In each iteration, every subproblem is solved to obtain a closed-form solution leading to a consistent increase in the objective function of problem (6). After denoting $\mathcal{L}(\boldsymbol{\alpha}, \mathbf{Y}) = \sum_{v_1=1}^{m_1} \alpha_{v_1} \mathbf{Tr}((\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{A}_{v_1}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}})$ and $\mathcal{L}(\boldsymbol{q}, \boldsymbol{p}, \mathbf{W}, \mathbf{Y}) = \sum_{v_2=1}^{m_2} q_{v_2} \|\hat{\mathbf{H}}_{v_2}\hat{\mathbf{H}}_{v_2}^T - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\|_F^2$, we have the following inequality:

$$\begin{aligned}&\mathcal{L}(\boldsymbol{\alpha}_{t+1}, \mathbf{Y}_{t+1}) - \lambda\mathcal{L}(\boldsymbol{q}_{t+1}, \boldsymbol{p}_{t+1}, \mathbf{W}_{t+1}, \mathbf{Y}_{t+1}) \\ &\qquad \geq \mathcal{L}(\boldsymbol{\alpha}_t \mathbf{Y}_t) - \lambda\mathcal{L}(\boldsymbol{q}_t, \boldsymbol{p}_t, \mathbf{W}_t, \mathbf{Y}_t) = 0.\end{aligned} \tag{21}$$

After a simple transformation, we have

$$\frac{\mathcal{L}(\boldsymbol{\alpha}_{t+1}\mathbf{Y}_{t+1})}{\mathcal{L}(\boldsymbol{q}_{t+1}, \boldsymbol{p}_{t+1}, \mathbf{W}_{t+1}, \mathbf{Y}_{t+1})} \geq \lambda = \frac{\mathcal{L}(\boldsymbol{\alpha}_t, \mathbf{Y}_t)}{\mathcal{L}(\boldsymbol{q}_t, \boldsymbol{p}_t, \mathbf{W}_t, \mathbf{Y}_t)} \tag{22}$$

which completes the proof. ■

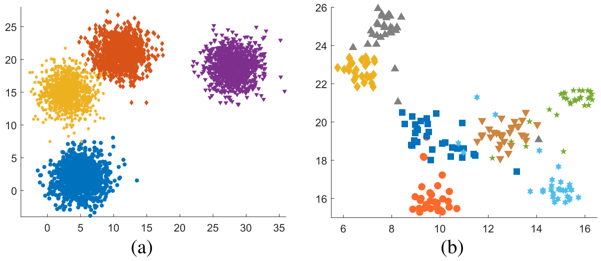
IV. Experiments

In this section, we use four real-world benchmark datasets, namely, Digit4k,[1] MSRC,[2] ORL,[3] and 100Leaves[4] to evaluate the effectiveness of CAGM.

TABLE I

Description of Datasets

Datasets	Class	Num.	View	d_1	d_2	d_3	d_4	d_5	d_6
Digit4k	4	4000	2	30	30	—	—	—	—
MSRC	7	210	6	1302	48	512	100	256	210
ORL	40	400	4	512	59	864	254	—	—
100Leaves	100	1600	3	64	64	64	—	—	—



(a) (b)

Fig. 2. T-SNE visualization results of cross-view embeddings $\hat{\mathbf{H}}_{v_2}$ on (a) Digit4k and (b) MSRC datasets.

The Digit4k dataset consists of 4000 digit images divided into four classes. Two sets of features are collected from Census Bureau employees and high-school students. The MSRC dataset encompasses 210 images allocated among seven distinct classes. Associated with each image are quintuple eigenvectors: color moment, gist, cent, hog, and LBP. And the ORL dataset aggregates images of human faces into 40 classes, culminating in a total of 400 images. Each image within this dataset is distinguished by a trio of attributes: a 512-D gist descriptor, a 59-D LBP descriptor, and 254 central features. The dataset named 100Leaves includes 1600 samples, each representing one of the 100 different plant species, and incorporates shape descriptors, fine-scale edges, and texture histograms. Table I presents the number of clusters (class), instances (num.), views, and the dimensionality of features in the vth view (d_v) in each dataset.

Six competitors for the state-of-the-art (SOTA) multiview clustering includes: AMGL [32], CGD [33], NESE [34], AWP [35], LSGMC [36], MMGC [37], and PFCEL [38]. Experiments are conducted on a workstation powered by an Intel Core i7-9700 CPU at 3.0 GHz, using MATLAB R2022b. During the preliminary setup, a K-nearest neighbors graph is constructed with a fixed number of nearest neighbors (k) set to 5 [39]. For postprocessing, spectral clustering is applied in CGD and LSGMC, whereas AMGL uses the K-means approach. To maintain equitable conditions, both spectral clustering and K-means run ten times with varied initial seeds averaging the outcomes as per [40]. Cluster analysis is quantified through four metrics: accuracy (ACC), normalized mutual information (NMI), purity, and adjusted rand index (ARI) with higher figures denoting superior results. Table II showcases the results with the top-performing results highlighted in bold and the second-best results underlined.

The proposed CAGM model consistently demonstrates optimal or near-optimal performance across the four datasets. For example, CAGM outperforms the SOTA by **0.0302** (ACC), **0.0152** (NMI), **0.0291** (purity), and **0.0305** (ARI) on the Digit4k dataset and **0.0190** (ACC), **0.0372** (NMI), **0.0190** (purity), and **0.0382** (ARI) on the MSRC dataset. Furthermore, we use the t-distributed stochastic neighbor embedding (T-SNE) algorithm [41] to visualize the separability of one cross-view embedding $\hat{\mathbf{H}}_{v_2} = \sum_{v_1=1}^{m_1} p_{v_1}\varphi(\mathbf{A}_{v_1})\mathbf{X}_{v_2}\mathbf{W}_{v_2}$ on the Digit4k and MSRC datasets, as shown in Fig. 2.

The convergence curves on four real-world datasets are illustrated in Fig. 3. Thanks to the distinguishability achieved by the learned embeddings, clustering is achieved in a few iterations. Fig. 4 demonstrates that the convergence speed of adjusting weights $\boldsymbol{\alpha}$, $\boldsymbol{p}$, and $\boldsymbol{q}$ also becomes stable rapidly. Due to minor variations in $\boldsymbol{p}$

[1] http://yann.lecun.com/exdb/mnist/

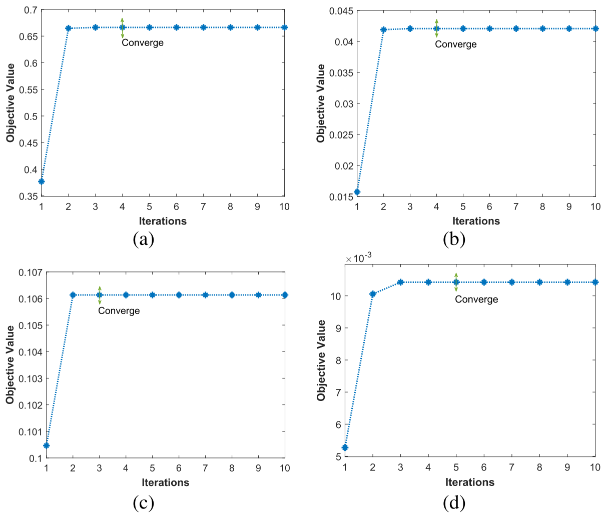
[2] https://mldta.com/dataset/msrc-v1/

[3] http://www.U.K..research.att.com/facedatabase.html

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

TABLE II
CLUSTERING PERFORMANCE ON FOUR REAL-WORLD DATASETS

Datasets	Methods	ACC	NMI	Purity	ARI
Digit4k	AMGL	0.6483	0.6496	0.7403	0.5774
	CGD	0.6430	0.7368	0.5692	0.5692
	NESE	0.8088	0.6569	0.8088	0.6224
	AWP	0.6488	0.6646	0.7443	0.5867
	LSGMC	0.6310	0.5862	0.7188	0.5135
	MMGC	0.6388	0.5961	0.7295	0.5507
	PFCEL	0.6478	0.6506	0.7408	0.5787
	CAGM	**0.8390**	**0.6798**	**0.8390**	**0.6529**
MSRC	AMGL	0.8476	0.7544	0.8476	0.6759
	CGD	0.8048	0.7234	0.8048	0.6151
	NESE	0.8810	0.7958	0.8810	0.7486
	AWP	0.8000	0.7383	0.8095	0.6673
	LSGMC	0.6310	0.5862	0.7188	0.5135
	MMGC	0.9048	0.8231	0.9048	0.7849
	PFCEL	0.8667	0.7490	0.8667	0.7150
	CAGM	**0.9238**	**0.8603**	**0.9238**	**0.8231**
ORL	AMGL	0.7825	0.9128	0.8425	0.6998
	CGD	0.7700	0.8943	0.8200	0.6140
	NESE	0.8225	0.9179	0.8600	0.7713
	AWP	0.7775	0.9096	0.8100	0.7453
	LSGMC	**0.8525**	0.9233	0.8775	0.7996
	MMGC	**0.8525**	0.9316	0.8775	**0.8093**
	PFCEL	0.7925	0.9233	0.8500	0.7643
	CAGM	**0.8525**	**0.9359**	**0.8900**	0.8052
100Leaves	AMGL	0.7725	0.8821	0.8006	0.5942
	CGD	0.5000	0.7250	0.5431	0.3305
	NESE	0.8196	0.8997	0.8456	0.7416
	AWP	0.7569	0.8781	0.7819	0.6533
	LSGMC	**0.8331**	**0.9194**	0.8431	**0.7695**
	MMGC	0.4775	0.7055	0.5069	0.3174
	PFCEL	0.7144	0.8586	0.7563	0.5161
	CAGM	**0.8331**	0.9025	**0.8525**	0.6001

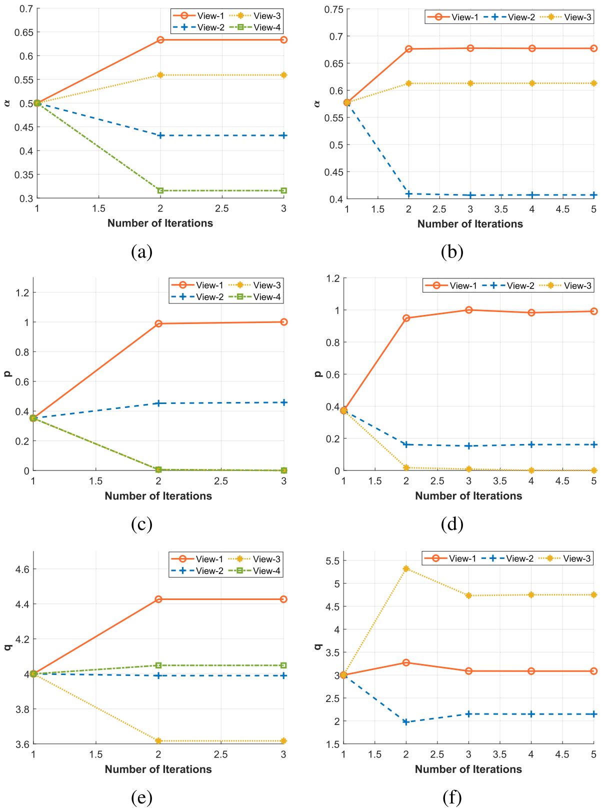


(a) (b) (c) (d)

Fig. 3. Convergence curves on four real-world datasets. (a) Digit4k, (b) MSRC, (c) ORL, and (d) 100Leaves.

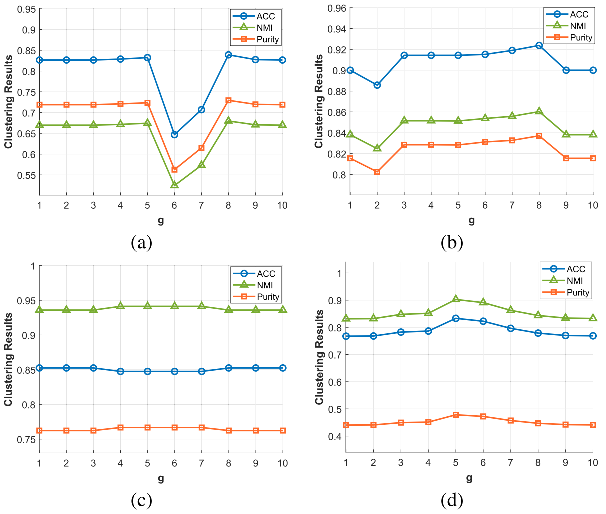
during updating, normalization adjustments are applied to the vertical coordinate settings.

Furthermore, we explore the influence of the graph filter order g on metrics: ACC, NMI, and purity. This investigation is illustrated in Fig. 5, where the parameter g is varied from 1 to 10 across four real-world datasets. Herein, the clustering performance demonstrates fluctuations with varying values of g, reflecting both enhancements and deteriorations. As g increases, information propagation expands, resulting in smoother feature updates. Fluctuations observed in the metric curves can be attributed to the underfitting tendency associated with small g values, while larger g values lead to overfitting. The optimal value of g holds the potential to achieve a more balanced aggregation of multiview graph structures and features. Besides, the number of nearest neighbors k during graph construction has a substantial influence on the performance of relevant methods, such as AMGL, NESE, AWP, MEA, and our proposed method. Selecting an appropriate k allows for a finer equilibrium between local and global multiview information.



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

Fig. 4. Change in weights $\boldsymbol{\alpha}$, $\boldsymbol{p}$, and $\boldsymbol{q}$ on (a), (c), and (e) ORL and (b), (d), and (f) 100Leaves datasets.



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

Fig. 5. Performance metrics (ACC, NMI, purity) at varied graph filter order g on four real-world datasets. (a) Digit4k, (b) MSRC, (c) ORL, and (d) 100Leaves.

V. Conclusion

In this brief, we address the gap in the integration of multiple graph and feature spaces in multiview clustering by proposing a CAGM model. CAGM integrates a paired encoding mechanism aimed at producing low-dimensional and orthogonal cross-view embeddings derived from both multiview graph structures and features. Subsequently, CAGM uses a ratio-formed objective function and leverages the Grassmann manifold subspace approximation to directly generate the indicator matrix. The corresponding algorithm is derived to tackle this issue, and the experimental results across various real-world datasets showcase the efficiency of CAGM. Our future emphasis will be on optimizing the incomplete utilization of cross-view combinations from multiple graph and feature spaces.

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] M.-S. Chen et al., “Representation learning in multi-view clustering: A literature review,” *Data Sci. Eng.*, vol. 7, no. 3, pp. 225–241, Sep. 2022.

[3] S. Huang, Z. Kang, and Z. Xu, “Auto-weighted multi-view clustering via deep matrix decomposition,” *Pattern Recognit.*, vol. 97, Jan. 2020, Art. no. 107015.

[4] Q. Gao, P. Zhang, W. Xia, D. Xie, X. Gao, and D. Tao, “Enhanced tensor RPCA and its application,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2133–2140, Jun. 2021.

[5] W. Xia, T. Wang, Q. Gao, M. Yang, and X. Gao, “Graph embedding contrastive multi-modal representation learning for clustering,” *IEEE Trans. Image Process.*, vol. 32, pp. 1170–1183, 2023.

[6] Y. Yun, J. Li, Q. Gao, M. Yang, and X. Gao, “Low-rank discrete multi-view spectral clustering,” *Neural Netw.*, vol. 166, pp. 137–147, Sep. 2023.

[7] Z. Huang, J. T. Zhou, X. Peng, C. Zhang, H. Zhu, and J. Lv, “Multi-view spectral clustering network,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, vol. 2, no. 3, p. 4.

[8] H. Lu, Q. Gao, Q. Wang, M. Yang, and W. Xia, “Centerless multi-view k-means based on the adjacency matrix,” in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 7, pp. 8949–8956.

[9] H. Yang, Q. Gao, W. Xia, M. Yang, and X. Gao, “Multiview spectral clustering with bipartite graph,” *IEEE Trans. Image Process.*, vol. 31, pp. 3591–3605, 2022.

[10] 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. 2020.

[11] M. Chen and X. Li, “Entropy minimizing matrix factorization,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9209–9222, Nov. 2023.

[12] W. Xia, Q. Gao, Q. Wang, X. Gao, C. Ding, and D. Tao, “Tensorized bipartite graph learning for multi-view clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5187–5202, Apr. 2022.

[13] J. Xu et al., “Self-supervised discriminative feature learning for deep multi-view clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 7470–7482, Jul. 2023.

[14] Z. Chen, P. Lin, Z. Chen, D. Ye, and S. Wang, “Diversity embedding deep matrix factorization for multi-view clustering,” *Inf. Sci.*, vol. 610, pp. 114–125, Sep. 2022.

[15] J. Liu et al., “One-pass multi-view clustering for large-scale data,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 12344–12353.

[16] M. Xu, L. Qin, W. Chen, S. Pu, and L. Zhang, “Multi-view adversarial discriminator: Mine the non-causal factors for object detection in unseen domains,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 8103–8112.

[17] M. Chen, M. Gong, and X. Li, “Feature weighted non-negative matrix factorization,” *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 1093–1105, Feb. 2023.

[18] J. Wang, C. Tang, K. Sun, Z. Wan, W. Zhang, and A. Zomaya, “Efficient and effective one-step multiview clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 14, 2023, doi: 10.1109/TNNLS.2023.3253246.

[19] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, and E. Zhu, “Unified one-step multi-view spectral clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6449–6460, Jun. 2023.

[20] J. Wang et al., “Fast approximated multiple kernel K-means,” *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 13, 2023, doi: 10.1109/TKDE.2023.3340743.

[21] X. Liu, “Hyperparameter-free localized simple multiple lernel k-means with global optimum,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 8566–8576, Jul. 2023.

[22] Z. Kang et al., “Multi-graph fusion for multi-view spectral clustering,” *Knowl.-Based Syst.*, vol. 189, Feb. 2020, Art. no. 105102.

[23] 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.*, vol. 35, no. 11, pp. 11388–11402, Nov. 2023.

[24] Z. Lin, Z. Kang, L. Zhang, and L. Tian, “Multi-view attributed graph clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1872–1880, Feb. 2023.

[25] W. Yan, J. Xu, J. Liu, G. Yue, and C. Tang, “Bipartite graph-based discriminative feature learning for multi-view clustering,” in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 3403–3411.

[26] D. Wu, J. Lu, F. Nie, R. Wang, and Y. Yuan, “EMGC2f: Efficient multi-view graph clustering with comprehensive fusion,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2022, pp. 3566–3572.

[27] W. Rong et al., “Learning a consensus affinity matrix for multi-view clustering via subspaces merging on Grassmann manifold,” *Inf. Sci.*, vol. 547, pp. 68–87, Feb. 2021.

[28] Y. Jia, F. Nie, and C. Zhang, “Trace ratio problem revisited,” *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 729–735, Apr. 2009.

[29] Z. Wang, F. Nie, L. Tian, R. Wang, and X. Li, “Discriminative feature selection via a structured sparse subspace learning module,” in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 3009–3015.

[30] J. M. Steele, *The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[31] D. Wu et al., “GSPL: A succinct kernel model for group-sparse projections learning of multiview data,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 3185–3191.

[32] F. Nie, J. Li, and X. Li, “Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification,” in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 9, 2016, pp. 1–7.

[33] C. Tang et al., “CGD: Multi-view clustering via cross-view graph diffusion,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 5924–5931.

[34] 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.

[35] F. Nie, L. Tian, and X. Li, “Multiview clustering via adaptively weighted procrustes,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2022–2030.

[36] W. Lan et al., “Multiview subspace clustering via low-rank symmetric affinity graph,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 31, 2023, doi: 10.1109/TNNLS.2023.3260258.

[37] Y. Tan, Y. Liu, H. Wu, J. Lv, and S. Huang, “Metric multi-view graph clustering,” in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 8, pp. 9962–9970.

[38] D. Wu, F. Nie, X. Dong, R. Wang, and X. Li, “Parameter-free consensus embedding learning for multiview graph-based clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7944–7950, Dec. 2022.

[39] F. Nie, X. Wang, M. Jordan, and H. Huang, “The constrained Laplacian rank algorithm for graph-based clustering,” in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 1–8.

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

[41] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.