

Priori Anchor Labels Supervised Scalable Multi-View Bipartite Graph Clustering

Jiali You¹, Zhenwen Ren^{1,2,3*}, Xiaojian You¹, Haoran Li^{1,4}, Yuancheng Yao¹

¹School of National Defense Science and Technology, Southwest University of Science and Technology, Mianyang, China

²Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai, China

³SongShan Laboratory, Henan, China

⁴Department of School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen, China
yj11015004@163.com, rzw@njust.edu.cn, youxjswust@163.com, haoranli50@gmail.com, yao08072022@163.com

Abstract

Although multi-view clustering (MVC) has achieved remarkable performance by integrating the complementary information of views, it is inefficient when facing scalable data. Proverbially, anchor strategy can mitigate such a challenge a certain extent. However, the unsupervised dynamic strategy usually cannot obtain the optimal anchors for MVC. The main reasons are that it does not consider the fairness of different views and lacks the priori supervised guidance. To completely solve these problems, we first propose the priori anchor graph regularization (PAGG) for scalable multi-view bipartite graph clustering, dubbed as SMGC method. Specifically, SMGC learns a few representative consensus anchors to simulate the numerous view data well, and constructs a bipartite graph to bridge the affinities between the anchors and original data points. In order to largely improve the quality of anchors, PAGG predefines prior anchor labels to constrain the anchors with discriminative cluster structure and fair view allocation, such that a better bipartite graph can be obtained for fast clustering. Experimentally, abundant of experiments are accomplished on six scalable benchmark datasets, and the experimental results fully demonstrate the effectiveness and efficiency of our SMGC.

Introduction

Multi-view clustering (MVC) has been widely researched, because it can integrate the diversity and complementary information between views. In previous studies, researchers have proposed abundant MVC methods, among which subspace learning is one of the most attractive research topics. MVC methods based on subspace learning usually divide into two steps, *i.e.*, first learn the self-representation matrix, and then perform spectral clustering. For the first step, if there are n samples from k clusters, the existing methods construct an affinity graph (*i.e.*, $\mathbf{Z} \in \mathbb{R}^{n \times n}$) by learning the similarity relationship between sample pairs. However, the computational complexity of optimizing \mathbf{Z} usually costs $\mathcal{O}(n^3)$ (Wang and Wu 2018; Wang et al. 2019; Zhang et al. 2020). For the second step, they perform the singular value decomposition (SVD) on the Laplacian matrix of \mathbf{Z} to obtain the clustering indicator, which costs high compu-

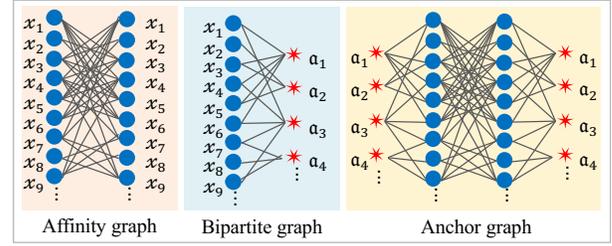


Figure 1: The schematic of different graphs. The first one is the affinity graph $\mathbf{Z} \in \mathbb{R}^{n \times n}$ of n samples used in the traditional MVC methods, costing $\mathcal{O}(n^3)$ computational complexity; the second one is the bipartite graph $\mathbf{Z} \in \mathbb{R}^{n \times m}$ between n samples and m ($m \ll n$) anchors used in traditional MVC methods for handling scalable data; and the third one is the anchor graph $\mathbf{Z} \in \mathbb{R}^{m \times m}$ of m anchors proposed in our SMGC.

tational complexity of $\mathcal{O}(n^3)$. Hence, most of the existing MVC methods are not suitable for scalable datasets.

To address this issue, MVC methods based on anchor are proposed in recent years, which first choose a few anchors, and then learn a bipartite graph, as shown in Fig. 1. For producing anchors, existing anchor strategies can be divided into static selection and dynamic learning. In the existing literature, the commonly used static selection strategy includes random sampling and k -means selection, which are performed independently before clustering (Yang et al. 2022a; Qiang et al. 2021; Yang et al. 2022b). Although the above methods can alleviate the problem of computational complexity, however anchors may not be able to simulate the original data points well. In order to improve the quality of the anchors, dynamic anchor learning is proposed, which makes the anchors with flexibility and presentation (Ou et al. 2020; Zhou et al. 2021; You et al. 2022). However, dynamic anchor learning is unsupervised, which does not consider the fairness and uniformity of anchors, resulting in **non-uniform sampling (NUS)** problem. That is, these anchors learning strategies may result in a non-uniform number of anchors selected in each cluster, or even no anchors selected in some clusters. As we know, anchors are the key to learn bipartite graph for fast clustering. Therefore, the NUS problem will affect the quality of the bipartite graph, resulting in

*Corresponding author

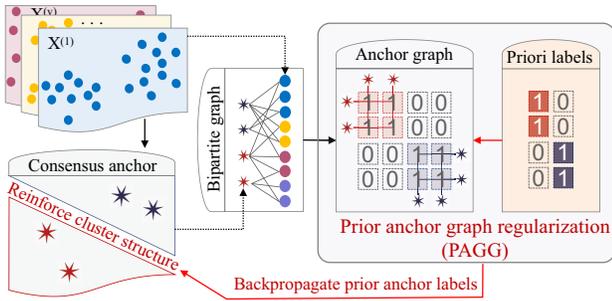


Figure 2: The schematic of the proposed SMGC, which aims to use optimal supervised anchors to handle scalable multi-view datasets. PAGG uses the prior anchor labels (PAL) to directly supervise anchor graph, and then backpropagate to the consensus anchor. Based on this, we constrain the anchor with discriminative cluster structure and fair view allocation, and then obtain a better bipartite graph for fast clustering.

suboptimal clustering performance.

In order to solve the above problem, we propose a novel multi-view bipartite graph clustering (SMGC) method, which aims to learn optimal anchors relying on the proposed the priori anchor graph regularization (PAGG). As shown in Fig. 2, firstly, we innovatively proposed the prior anchor graph regularization (PAGG), which utilizes the priori anchor labels (PAL) to constrain anchors have fairer and more discriminative cluster structure. To be detail, we use PAL to directly constrain the anchor graph to have clear cluster structure, and then indirectly guide the anchors with uniform distribution. Based on this, we learn the similarity relationship between the learned anchors and the original data points, so that we can achieve a high quality bipartite graph. Finally, the clustering indicator is obtained by singular value decomposition (SVD) on the bipartite graph, and then it is discretized by k -means algorithm. Compared to previous MVC methods based on anchor, the main contributions of our SMGC are summarized as follows:

- The priori anchor graph regularization (PAGG) is innovatively proposed, which effectively solve the NUS problem. In detail, PAGG utilizes prior anchor labels (PAL) to constrain the anchors with discriminative cluster structure and fair view allocation.
- A novel method, *i.e.*, SMGC is proposed for fast clustering, which can efficient handle scalable datasets. Specifically, under the guidance of PAGG, we integrate anchor selection and bipartite graph learning into a framework to collaboratively optimize the consensus anchor.
- Plentiful experiments are carried out on six benchmark datasets, and the effectiveness and efficiency of the proposed SMGC are verified. Moreover, we visualized the anchors distribution of the proposed SMGC, which fully proves the feasibility of PAGG.

Related Work

In this section, the notations are summarized, and then bipartite graph learning and kernel k -means are reviewed.

| Notation | Description |
|--|--|
| d_i | The number of features. |
| n | The number of samples. |
| k | The number of clusters. |
| l | The number of anchors. |
| $\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$ | Data matrix for the i -th view. |
| $\mathbf{Z} \in \mathbb{R}^{n \times n}$ | Affinity matrix of the all views. |
| $\mathbf{W}_i \in \mathbb{R}^{d_i \times l}$ | Projection matrix of the i -th view. |
| $\mathbf{A} \in \mathbb{R}^{l \times l}$ | Consensus anchor matrix for all views. |
| $\mathbf{R} \in \mathbb{R}^{l \times n}$ | Consensus bipartite graph for all views. |
| $\mathbf{F} \in \mathbb{R}^{l \times c}$ | Prior anchor graph indicator. |

Table 1: Notations used throughout paper.

Notations

To facilitate reading, we have summarized the notations involved among this article in Table 1. Moreover, we visualized the affinity graph, bipartite graph and anchor graph to help distinguish and understand, as shown in Fig. 1.

Bipartite Graph Clustering

Give a multi-view dataset $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_v] \in \mathbb{R}^{d_i \times n}$ consisting of v different views, where d_i is the number of features in the i -th view, and n denote the number of samples. For MVC, subspace clustering is a common method, which is based on the assumption that each sample can be represented by a linear combination of other samples. Mathematically, subspace clustering can be expressed as follows:

$$\min_{\mathbf{Z}} \Xi(\mathbf{X}, \mathbf{XZ}) + \rho \Psi(\mathbf{Z}) \quad (1)$$

where ρ is the tradeoff parameter, $\Xi(\cdot, \cdot)$ and $\Psi(\cdot)$ represent the loss function and regularization term, respectively. Meanwhile, $\mathbf{Z} \in \mathbb{R}^{n \times n}$ stands for affinity matrix, which essentially needs to calculate the similarity between n^2 sample pairs, as shown in Fig. 1. In order to reduce the computational complexity, bipartite graph clustering is proposed (Zhou et al. 2021; Liu et al. 2022), which is essentially to compress the information of the scalable dataset and then learn the potential features. That is, the mathematical expression of bipartite graph clustering is

$$\min_{\mathbf{R}, \mathbf{D}} \Xi(\mathbf{X}, \mathbf{DR}) + \rho \Psi(\mathbf{R}) \quad (2)$$

where $\mathbf{D} \in \mathbb{R}^{d_i \times l}$ denotes anchor matrix, l is the number of anchors, and $\mathbf{R} \in \mathbb{R}^{l \times n}$ is bipartite graph representing the similarity relationship between anchors and original data points. In MVC, Frobenius norm (*i.e.*, $\|\cdot\|_F$) is often used as the loss function. Thus, the first term is expressed as $\|\mathbf{X}_i - \mathbf{D}_i \mathbf{R}\|_F^2$, where \mathbf{D}_i represents the anchor matrix of the i -th view. In general, the anchors are the key to the quality of bipartite graphs (Sun et al. 2021).

Kernel k -means Clustering

The kernel k -means (KKM) is the most widely applied clustering method, whose goal is to minimize the sum of squares

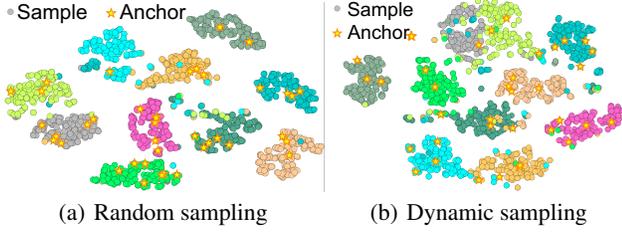


Figure 3: Visualization of anchors on the UCI-DIGIT dataset by random sampling and dynamic learning, respectively.

loss over the cluster assignment matrix $\mathbf{Y} \in \mathbb{R}^{n \times c}$. Mathematically, KKM can be expressed as

$$\min_{\mathbf{Y}} \sum_{i=1, c=1}^{n, k} \mathbf{Y}_{ic} \|\phi(\mathbf{x}_i) - \boldsymbol{\mu}_k\|_2^2 \quad \text{s.t. } \mathbf{Y}\mathbf{1}_k = \mathbf{1}_n \quad (3)$$

where $\boldsymbol{\mu}_k$ is the clustering center, $\mathbf{Y} \in \{0, 1\}^{n \times k}$ is a discrete clustering assignment, and $\phi(\cdot)$ is the non-linear projection that projects data in the original space to reproducing kernel Hilbert space (RKHS). Further, Eq. (3) can be simplified into the following matrix-vector form,

$$\min_{\mathbf{H} \in \mathbb{R}^{n \times k}} \text{Tr}(\mathbf{K}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)) \quad \text{s.t. } \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k \quad (4)$$

where $\mathbf{K} \in \mathbb{R}^{n \times n}$ is a kernel matrix, which each entry is the kernel of the corresponding data *i.e.*, $k_{ij} = \phi(x_i)^\top \phi(x_j)$. Further, considering that $\text{Tr}(\mathbf{K})$ is a constant term, so Eq. (4) can be simplified to

$$\min_{\mathbf{H} \in \mathbb{R}^{n \times k}} -\text{Tr}(\mathbf{H}^\top \mathbf{K} \mathbf{H}) \quad \text{s.t. } \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k \quad (5)$$

It is well known that kernel matrix has the following properties, (1) kernel matrix is a symmetric matrix. For the non-linear kernel, it is the inner product in a feature space with feature map *i.e.*, $k_{ij} = \phi(x_i)^\top \phi(x_j)$ (Ng 2000), and (2) in general, kernel matrix is $\mathbf{K}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, for $i, j = 1, \dots, \ell$. For any vector, we have $\mathbf{v}^\top \mathbf{K} \mathbf{v} = \sum_{i,j=1}^{\ell} v_i v_j \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \|\sum_{i=1}^{\ell} v_i \phi(\mathbf{x}_i)\|^2 \geq 0$, so kernel matrix is a positive semi-definite matrix (Lemmetty, Keskinen, and Marjamäki 2005). Overall, kernel matrix can be deemed as affinity graph with symmetric and positive semi-definite (SPSD) constraints.

Proposed Method

In this section, we find that existing anchor strategies lead to the non-uniform sampling (NUS) problem. Then, we propose priori anchor labels supervised scalable multi-view bipartite graph clustering (SMGC) to solve this problem.

Problem Finding

When faced with scalable multi-view datasets, MVC methods based on anchor are widely used for fast clustering. In general, the quality of the anchors are important, which directly determines the quality of the bipartite graph, and indirectly affects the clustering performance (Lu et al. 2022).

Thus, various anchor strategies are proposed to select few anchors. However, through the visualization experiment of anchors, it is found that the existing anchor strategies are easy to lead anchors with non-uniform distribution, as shown in Fig. 3. From the Fig. 3, we can find that the number of anchors is non-uniform in each cluster, and even no anchors are selected in some clusters. This is because the existing anchor strategies are unsupervised and do not consider the cluster structure of anchors and fair viewpoint allocation. In summary, the existing anchor strategies have the problem of **non-uniform sampling (NUS)**.

Priori Anchor Graph Regularization (PAGG)

The NUS problem reduces the reliability of anchors and affects the quality of bipartite graph. To address this issue, we construct a priori anchor labels (PAL) *i.e.*, \mathbf{F} , which aim to constrain the anchors with discriminative cluster structure and fair view allocation by using a set of preset indicator vectors. For a column \mathbf{f}_j , which represents the clustering indicator of an anchor, if it belongs to the first cluster, then $\mathbf{f}_{1j} = 1$ and the other elements are 0.

Take PAL as the supervised information, we innovatively propose the **prior anchor graph regularization (PAGG)**, as shown in blue part in Fig. 2. To be specific, we consider the cluster structure of anchors and the uniformity of anchors in each cluster. Then, we assume that we have a bipartite graph $\mathbf{R} \in \mathbb{R}^{\ell \times n}$, and use it to construct a small anchor graph $\mathbf{R}\mathbf{R}^\top \in \mathbb{R}^{\ell \times \ell}$. It is important to note that we treat the anchor graph as a kernel matrix rather than an ordinary affinity graph. This is because the anchor graph $\mathbf{R}\mathbf{R}^\top$ satisfies symmetry and positive semi-definite (SPSD), which is an ideal kernel matrix. Therefore, inspired by kernel k -means (KKM), PAGG can be mathematically expressed as

$$\min_{\mathbf{R}} -\text{tr}(\mathbf{F}^\top \mathbf{R}\mathbf{R}^\top \mathbf{F}) \quad \text{s.t. } \mathbf{R} \geq 0, \mathbf{R}^\top \mathbf{1} = \mathbf{1} \quad (6)$$

where $\mathbf{F} \in \mathbb{R}^{\ell \times c}$ is the priori anchor labels. Unlike Eq. (4), the clustering indicator of Eq. (6) is given in advance. In general, PAGG utilizes the PAL (\mathbf{F}) to directly supervise anchor graph, and then backpropagate to the anchor.

Proposed Formula

It is well known that anchors are the key to learning bipartite graph for fast clustering. Hence, we use PAGG to constrain the selection of anchors, and then learn the similarity relationship between optimal anchors and original data points to construct a bipartite graph. For bipartite graph learning, Eq. (2) is one of the most popular methods, because anchors are selected dynamically and have flexibility. However, Eq. (2) select anchors for each view individually, not taking into account that anchors for different views should have potential anchor representations. Thus, we focus on consensus anchors learning rather than anchors learning for each view, which is mathematically expressed as follows

$$\min_{\delta_i, \mathbf{W}_i, \mathbf{A}, \mathbf{R}} \sum_{i=1}^v \delta_i^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{R}\|_F^2 \quad \text{s.t. } \delta^\top \mathbf{1} = 1, \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_k, \mathbf{A}^\top \mathbf{A} = \mathbf{I}_k, \mathbf{R} \geq 0, \mathbf{R}^\top \mathbf{1} = \mathbf{1} \quad (7)$$

where δ_i is the weight of the i -th view, \mathbf{W}_i is the projection matrix, \mathbf{A} is the consensus anchor matrix, and \mathbf{R} is the consensus bipartite graph. Constraint $\mathbf{A}^\top \mathbf{A} = \mathbf{I}_k$ ensures consensus anchor matrix \mathbf{A} more discriminative.

Further, in order to obtain optimal anchors, we combine the proposed PAGG (*i.e.*, Eq. (7)) and Eq. (6) into a unified framework. Moreover, we name this method as priori anchor labels supervised scalable multi-view bipartite graph clustering (SMGC), which is mathematically expressed as

$$\begin{aligned} \min_{\delta_i, \mathbf{W}_i, \mathbf{A}, \mathbf{R}} \sum_{i=1}^v \delta_i^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{R}\|_F^2 - \beta \text{tr}(\mathbf{F}^\top \mathbf{R} \mathbf{R}^\top \mathbf{F}) \\ \text{s.t. } \delta^\top \mathbf{1} = 1, \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_k, \mathbf{A}^\top \mathbf{A} = \mathbf{I}_k, \mathbf{R} \geq 0, \mathbf{R}^\top \mathbf{1} = 1 \end{aligned} \quad (8)$$

where β is a regularization parameter used to control the balance of the PAGG. In general, the advantages of the proposed SMGC are as follows:

- The PAGG effectively solves the NUS problem, that is, it can make the anchors with discriminative cluster structure and fair view allocation. In essence, PAGG has the potential to be generalized and can become a necessary regularization term for anchor selection;
- It considers that the kernel matrix can be regarded as the affinity graph with SPSD constraints, and cleverly used this property in the design of PAGG;
- The uniform anchors guarantee a high-quality bipartite graph, resulting in optimal clustering performance via singular value decomposition (SVD).

Optimization

In this section, algorithm 1 is proposed to solve Eq. (8) effectively. At the same time, the memory complexity, computational complexity and convergence are analyzed in detail.

Solver for the Proposed Method

The Eq.(8) is not-convex. For a concise solution, we use an alternating iterative algorithm, updating one variable while fixing the others.

► **Update δ_i :** Optimization δ_i with fixed \mathbf{W}_i , \mathbf{A} and \mathbf{R} . Thus, the objective in Eq.(8) is formulated as

$$\min_{\delta_i} \sum_{i=1}^v \delta_i^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{R}\|_F^2 \quad \text{s.t. } \delta^\top \mathbf{1} = 1 \quad (9)$$

Letting $\mathbf{d}_i = \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{R}\|_F$, Eq.(9) can be reduced to,

$$\min_{\delta_i} \sum_{i=1}^v \delta_i^2 \mathbf{d}_i^2 \quad \text{s.t. } \delta^\top \mathbf{1} = 1 \quad (10)$$

Eq.(10) can be solved via Cauchy-Schwarz inequality (Wu et al. 2022).

$$\delta_i = \frac{\frac{1}{d_i}}{\sum_{v=1}^V \frac{1}{d_i}} \quad (11)$$

By solving Eq. (11), so we can obtain the optimal δ_i .

► **Update \mathbf{W}_i :** With δ_i , \mathbf{A} , and \mathbf{R} are fixed, we only need to minimize the following objective

$$\min_{\mathbf{W}_i} \sum_{i=1}^v \delta_i^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{R}\|_F^2 \quad \text{s.t. } \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_k \quad (12)$$

Letting $\mathbf{B}_i = \sum_{i=1}^V \delta_i^2 \mathbf{X}_i \mathbf{R}^\top \mathbf{A}^\top$, the Eq. (12) can be written as

$$\max_{\mathbf{W}_i} \text{Tr}(\mathbf{W}_i^\top \mathbf{B}_i) \quad \text{s.t. } \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_k \quad (13)$$

Fortunately, such problem can be solved by Theorem 1 (You et al. 2022).

Theorem 1. Let the SVD of \mathbf{B} be $\mathbf{B} = \mathbf{U} \mathbf{D} \mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{d \times l}$, $\mathbf{D} \in \mathbb{R}^{l \times l}$ and $\mathbf{V} \in \mathbb{R}^{l \times l}$. The optimal solution of problem $\max_{\mathbf{W}^\top \mathbf{W} = \mathbf{I}} \text{Tr}(\mathbf{W}^\top \mathbf{B})$ is given by $\mathbf{W}^* = \mathbf{U} \mathbf{V}^\top$.

Proof. Due to $\mathbf{B} = \mathbf{U} \mathbf{D} \mathbf{V}^\top \in \mathbb{R}^{d \times l}$, problem $\text{Tr}(\mathbf{W}^\top \mathbf{B})$ can be reformulated as

$$\text{Tr}(\mathbf{W}^\top \mathbf{B}) = \text{Tr}(\mathbf{W}^\top \mathbf{U} \mathbf{D} \mathbf{V}^\top) = \text{Tr}(\mathbf{D} \mathbf{V}^\top \mathbf{W}^\top \mathbf{U})$$

Letting $\mathbf{L} = \mathbf{V}^\top \mathbf{W}^\top \mathbf{U}$, d_{ii} and l_{ii} represent the i -th diagonal elements of matrix \mathbf{D} and \mathbf{L} , respectively. Considering $\mathbf{L}^\top \mathbf{L} = \mathbf{I}_k$, we have $|l_{ii}| \leq 1$; moreover, have $d_{ii} \geq 0$ since it is the singular value of matrix \mathbf{A} . Therefore,

$$\text{Tr}(\mathbf{W}^\top \mathbf{B}) = \text{Tr}(\mathbf{D} \mathbf{L}) = \sum_{i=1}^l d_{ii} l_{ii} \leq \sum_{i=1}^l d_{ii}$$

According to the above inequality, $\text{Tr}(\mathbf{W}^\top \mathbf{B})$ reaches to maximization when $l_{ii} = 1$. Thus $\mathbf{L} = [\mathbf{I}_d, 0] \in \mathbb{R}^{l \times n}$ and $\mathbf{L} = \mathbf{V}^\top \mathbf{W}^\top \mathbf{U}$, so $\mathbf{W} = \mathbf{U} [\mathbf{I}_d; 0] \mathbf{V}^\top$ can be obtained. Further, the solution of $\max_{\mathbf{W}^\top \mathbf{W} = \mathbf{I}} \text{Tr}(\mathbf{W}^\top \mathbf{B})$ can be rewritten as $\mathbf{W} = \mathbf{U} \mathbf{V}^\top$ based on the SVD of matrix \mathbf{B} . Hereto, the proof is completed. ■

► **Update \mathbf{A} :** With δ_i , \mathbf{W}_i , and \mathbf{R} are fixed, we only need to minimize the following objective

$$\min_{\mathbf{A}} \sum_{i=1}^v \delta_i^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{R}\|_F^2 \quad \text{s.t. } \mathbf{A}^\top \mathbf{A} = \mathbf{I}_k \quad (14)$$

which is equivalent to:

$$\max_{\mathbf{A}} \text{Tr}(\mathbf{A}^\top \mathbf{E}) \quad \text{s.t. } \mathbf{A}^\top \mathbf{A} = \mathbf{I}_k \quad (15)$$

where $\mathbf{E} = \sum_{i=1}^V \delta_i^2 \mathbf{W}_i^\top \mathbf{X}_i \mathbf{R}^\top$. Similar to Eq. (13), the optimal \mathbf{A} can be solved by Theorem 1.

► **Update \mathbf{R} :** Fixing the other variables except \mathbf{Z} , the optimal problem can be reduced to

$$\begin{aligned} \min_{\mathbf{R}} \sum_{i=1}^v \delta_i^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{R}\|_F^2 - \beta \text{tr}(\mathbf{F}^\top \mathbf{R} \mathbf{R}^\top \mathbf{F}) \\ \text{s.t. } \mathbf{R} \geq 0, \mathbf{R}^\top \mathbf{1} = 1 \end{aligned} \quad (16)$$

It is equivalent to solving the following Quadratic Programming (QP) problem (Wang et al. 2022).

$$\begin{aligned} \min \frac{1}{2} \mathbf{R}_{:,j}^\top \mathbf{M} \mathbf{R}_{:,j} + n^\top \mathbf{R}_{:,j} \\ \text{s.t. } \mathbf{R}_{:,j}^\top \mathbf{1} = 1, \mathbf{R} \geq 0 \end{aligned} \quad (17)$$

where $\mathbf{M} = 2((\sum_{i=1}^V \delta_i^2) \mathbf{I} - \beta \mathbf{F} \mathbf{F}^\top)$ and $\mathbf{N} = -2 \sum_{i=1}^V \mathbf{X}_{:,j}^\top \mathbf{W}_i \mathbf{A}$.

In sum, we summarize Algorithm 1 for solving Eq. (8).

Algorithm 1 The algorithm of SMGC

Input: The multi-view data $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_v] \in \mathbb{R}^{d_i \times n}$, the regularization parameter β , and the priori anchors indicator $\mathbf{F} \in \mathbb{R}^{l \times c}$.

Initialize: Initialize \mathbf{W} , \mathbf{A} , and \mathbf{R} is a identity matrix with dimensions $d_i \times l$, $l \times l$ and $l \times n$ respectively, and $\delta_i = \frac{1}{v}$.

Output: Perform k -means on bipartite matrix \mathbf{R} to obtain cluster indicator.

- 1: **while** condition **do**
 - 2: Solve weight δ_i by Eq. (10).
 - 3: Solve projection matrix \mathbf{W}_i by Eq. (13).
 - 4: Solve consensus anchor matrix \mathbf{A} by Eq. (15).
 - 5: Solve consensus bipartite graph \mathbf{R} by Eq. (17).
 - 6: **end while**
-

Complexity Analysis

In this section, we analyze the memory complexity and time complexity of the proposed SMGC.

Memory complexity: The proposed SMGC needs to store three matrix variables, namely $\mathbf{W}_i \in \mathbb{R}^{d_i \times l}$, $\mathbf{A} \in \mathbb{R}^{l \times l}$ and $\mathbf{R}^{l \times n}$, with memory complexity of $O(d_i l)$, $O(l^2)$, and $O(ln)$, respectively. Therefore, the total memory complexity is $O(d_i l + l^2 + ln) \approx O(n)$, because $d_i \ll n$ and $l \ll n$.

Time complexity: In this paper, time complexity is caused by the optimization of four sub-problems. For the δ_i sub-problem, only $O(1)$ of computational complexity is require. For the \mathbf{W}_i and \mathbf{A} sub-problem, the main time complexity cost is singular value decomposition (SVD) of $\mathbf{B} \in \mathbb{R}^{d_i \times l}$ and $\mathbf{E} \in \mathbb{R}^{l \times l}$, respectively. Therefore, their time complexity is $O(d_i l^2)$ and $O(l^3)$. For the \mathbf{R} sub-problems, this is a QP problem, which the time complexity is $O(nl^3)$ for all columns. Thus, the total time complexity is $O(1 + d_i l^2 + (n + 1)l^3) \approx O(n)$.

Comparison and analysis: Furthermore, in order to demonstrate the efficiency of SMGC, we compare other methods and summarize them in Table 2. From the Table 2, it can be seen that the proposed method has lower memory (*i.e.*, $O(n)$) and computation complexity (*i.e.*, $O(n)$).

| Method | Time Complexity | Memory Complexity |
|----------|-----------------|------------------------|
| MLRSSC | $O(n^3)$ | $O((v + 1)n^2)$ |
| BMVC | $O(n)$ | $O(lm)$ |
| FMR | $O(n^3)$ | $O(n^2 + mn)$ |
| mPAC | $O(vmn^2)$ | $O(mn^2)$ |
| PMSC | $O(n^3)$ | $O(2vn^2 + (v + 1)nk)$ |
| MLES | $O(n^3)$ | $O(vn^2)$ |
| LMVSC | $O(n^4)$ | $O(vk(n + h))$ |
| SFMC | $O(nmd + nm^2)$ | – |
| CAG | $O(n)$ | $O(kn + (h + k)k)$ |
| SMVSC | $O(n)$ | $O(mn + (h + m)k)$ |
| Proposed | $O(n)$ | $O(n)$ |

Table 2: Summary of the time complexity and memory complexity of all comparison methods, where n , m , v and l respectively represent the number of samples, features, views and anchors, and $h = \sum_{i=1}^v d_i$.

Convergence Analysis

Theoretically, the objective function Eq. (8) is bounded and lower bounded, and convergence can be proved. Firstly, we omit these constraints, and then Eq. (8) can be converted to

$$\begin{aligned} & \min_{\delta_i, \mathbf{W}_i, \mathbf{A}, \mathbf{R}} \sum_{i=1}^v \delta_i^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{R}\|_F^2 - \beta \text{tr}(\mathbf{F}^\top \mathbf{R} \mathbf{R}^\top \mathbf{F}) \\ &= \min_{\delta_i, \mathbf{W}_i, \mathbf{A}, \mathbf{R}} \sum_{i=1}^v \delta_i^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{R}\|_F^2 + \beta \|\mathbf{F} \mathbf{F}^\top - \mathbf{R} \mathbf{R}^\top\|_F^2 \\ &= \min_{\delta_i, \mathbf{W}_i, \mathbf{A}, \mathbf{R}} \left\| \mathbf{R} - \frac{\sum_{i=1}^v \delta_i^2 (\mathbf{X}_i^\top \mathbf{W}_i \mathbf{A}) + \beta \mathbf{F}^\top}{\sum_{i=1}^v \delta_i^2 + \mathbf{I}} \right\|_F^2 \geq 0 \end{aligned} \quad (18)$$

According to the above analysis, the objective function *i.e.*, Eq. (8) has a lower bound of 0, so the convergence of algorithm 1 can be guaranteed.

Experiments

In this section, we verify the effectiveness and efficiency of the proposed SMGC through extensive of experiments.

Experiment Settings and Implementation Details

Benchmark datasets: We extensively evaluate the clustering performance of our proposed SMGC on six real multi-view benchmark datasets, including Caltech101-20 (Wu et al. 2022), CCV (Wang et al. 2021), Caltech101-all (Sun et al. 2021), SUNRGBD (Liu et al. 2021), NUSWIDE OBJ (Du et al. 2021), and AwA (Yang et al. 2022a). The number of samples, views, clusters and size of these datasets ranges from 2386 to 30475, 2 to 6, 20 to 120, and 3M to 543.4M, respectively. To more intuitively reflect the differences between datasets, we summarize the details in Table 3.

| Dataset | Samples | Views | Clusters | Size |
|----------------|---------|-------|----------|-------|
| | n | v | k | MB |
| Caltech101-20 | 2386 | 6 | 20 | 34.8 |
| CCV | 6773 | 3 | 20 | 3 |
| Caltech101-all | 9144 | 5 | 102 | 133.6 |
| SUNRGBD | 10335 | 2 | 45 | 229.5 |
| NMSWIDE OBJ | 30000 | 5 | 31 | 112.6 |
| AwA | 30475 | 6 | 50 | 543.4 |

Table 3: Summary of the used MVC benchmark datasets.

Benchmark methods: In order to verify the superiority of the proposed SMGC, we choose state-of-the-art methods from the last five years. To be more specific, we chose 10 methods as comparison methods, namely **MLRSSC** (Brbić and Kopriva 2018), **BMVC** (Zhang et al. 2018), **mPAC** (Kang et al. 2019), **PMSC** (Kang et al. 2020), **LMVSC** (Zhan et al. 2018), **SFMC** (Li et al. 2020), **FPMVC-CAG** or **CAG** (Wang et al. 2021), and **SMVSC** (Sun et al. 2021). These methods can be roughly divided into two categories. The first category is MVC methods based on anchor, *i.e.*, BMVC, SFMC, FPMVC-CAG, and SMVSC. The other competitors are MVC methods based on subspace learning.

| Dataset | Metrics | MLRSSC 2018 | BMVC 2018 | mPAC 2019 | PMSC 2020 | LMVSC 2020 | SFMC 2021 | CAG 2021 | SMVSC 2021 | Ours |
|----------------|---------|----------------|---------------|--------------|--------------|---------------|--------------|---------------|---------------|---------------|
| Caltech101-20 | ACC | 0.3600 | 0.1769 | 0.4893 | 0.5981 | 0.4304 | 0.5947 | 0.6547 | 0.6132 | 0.6697 |
| | NMI | 0.2008 | 0.1708 | 0.5855 | 0.5244 | 0.5553 | 0.5641 | 0.6326 | 0.5873 | 0.6423 |
| | PUR | 0.4476 | 0.4166 | 0.6622 | 0.6480 | 0.7125 | 0.7045 | 0.7368 | 0.6999 | 0.7485 |
| CCV | ACC | 0.1259 | 0.1326 | 0.2311 | — | 0.2014 | 0.1156 | 0.2399 | 0.2182 | 0.2368 |
| | NMI | 0.0471 | 0.0763 | 0.1744 | — | 0.1657 | 0.0346 | 0.1760 | 0.1684 | 0.1792 |
| | PUR | 0.1307 | 0.1652 | 0.2917 | — | 0.2396 | 0.1194 | 0.2605 | 0.2439 | 0.2622 |
| Caltech101-all | ACC | 0.1365 | 0.2123 | 0.2031 | — | 0.2005 | 0.1777 | 0.3015 | 0.2750 | 0.3584 |
| | NMI | 0.1066 | 0.4246 | 0.3809 | — | 0.4155 | 0.2613 | 0.3549 | 0.3510 | 0.3665 |
| | PUR | 0.1371 | 0.4124 | 0.2914 | — | 0.3975 | 0.2430 | 0.3460 | 0.3395 | 0.4056 |
| SUNRGBD | ACC | 0.1741 | 0.1669 | 0.1906 | — | 0.1858 | 0.1113 | 0.2392 | 0.1930 | 0.2462 |
| | NMI | 0.1108 | 0.1954 | 0.1335 | — | 0.2607 | 0.0202 | 0.2418 | 0.2007 | 0.2479 |
| | PUR | 0.1741 | 0.3357 | 0.1992 | — | 0.3818 | 0.1144 | 0.3400 | 0.2971 | 0.3526 |
| NUSWIDEOBJ | ACC | — | 0.1299 | — | — | 0.1583 | 0.1221 | 0.1946 | 0.1916 | 0.2210 |
| | NMI | — | 0.1290 | — | — | 0.1337 | 0.0095 | 0.1351 | 0.1272 | 0.1309 |
| | PUR | — | 0.2333 | — | — | 0.2488 | 0.1227 | 0.2382 | 0.2331 | 0.2406 |
| AwA | ACC | — | 0.0867 | — | — | 0.0770 | 0.0390 | 0.0919 | 0.0878 | 0.1006 |
| | NMI | — | 0.1195 | — | — | 0.0879 | 0.0032 | 0.1083 | 0.1061 | 0.1130 |
| | PUR | — | 0.1094 | — | — | 0.0957 | 0.0399 | 0.0961 | 0.0993 | 0.1172 |

Table 4: Clustering results of the comparison methods, where boldface indicates the optimal result. Moreover, the ‘—’ represents the result that cannot be obtained due to out-of-memory.

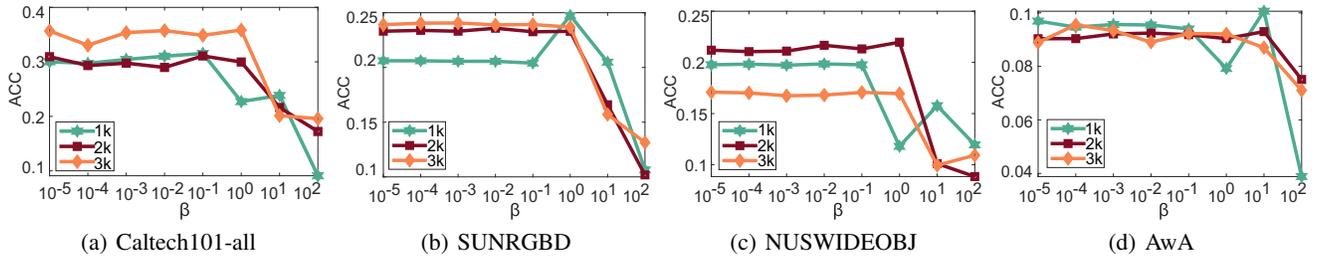


Figure 4: Parameter sensitivity of the proposed SMGC on four benchmark datasets.

Evaluating Metrics: In this paper, normalized mutual information (NMI), accuracy (ACC) and Purity (PUR) are used to evaluate the clustering performance of the proposed SMGC. Besides, in order to be fair, the comparison methods adopt the parameters recommended by the corresponding article. Considering the randomness of k -means due to random center initialization, we perform k -means 30 times to eliminate randomness.

Experimental Results and Analysis

In order to investigate the clustering performance of our proposed SMGC in MVC, we conducted experiments on 10 state-of-the-art methods on six benchmark datasets, and the results are shown in Table 4. Based on this table, we can draw the following conclusion.

- On the whole, the proposed SMGC achieves the best clustering performance on most datasets, which adequately demonstrates the feasibility of our SMGC.
- Compared with the traditional MVC methods based on subspace learning, the proposed SMGC achieves the best performance. This is because the process of anchors

learning can essentially eliminate a lot of noise and outliers, which is conducive to subsequent clustering. In other words, our anchors learning method is an effective way of selecting representative samples from the original data points, which makes the data relatively clean.

- Compared with the other four MVC methods based on anchors (*i.e.*, SFMC, CAG, SMVSC and ours), BMVC and SFMC have poor performance. The former dynamically selects the anchors from the original data points, so that the anchors are more flexible. However, the latter adopts the static anchor strategy, that is, the anchor is fixed and cannot change again. Thus, the advantage of dynamic anchor selection is proved effectively.
- Our proposed SMGC has better clustering performance than CAG and SMVSC, because we have effectively solved the NUS problem. The competitors do not consider the fairness of different views and lacks the priori supervised guidance. However, we innovatively proposes prior anchor graph regularization (PAGG) with using prior anchor labels (PAL) to directly supervise anchor graph, and then backpropagate to the consensus anchor.

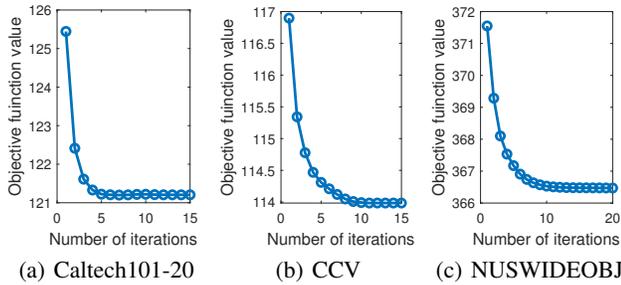


Figure 5: The objective value of the proposed SMGC.

Parameter Sensitivity and Convergence Analysis

In this section, we utilize the grid search to adjust parameter β with the range of $[10^{-5}, 10^{-4}, \dots, 10^1, 10^2]$, so as to study the influence of parameter in SMGC. In detail, we performed parameter sensitivity experiments on four benchmark datasets, and the results are shown in Fig. 4. We can draw following conclusions: (1) Our proposed SMGC only involves one parameter *i.e.*, β , which is easy to adjust; (2) When the values of β are not too large, *e.g.*, $[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$, the parameter β are relatively insensitive. Moreover, it is important that ACC achieves the best when $\beta = 1$ or $\beta = 10$, which proves the effectiveness of the proposed PAGG; and (3) Under the same β setting, the number of anchors will affect the clustering performance. Obviously, when the number of anchors are k or $2k$, most datasets have the best performance. This proves the importance of anchors selection.

Experimentally, we record the objective function value of the proposed SMGC in each iteration. Due to space constraints, we present the corresponding results of three datasets, *i.e.*, Caltech101-20, CCV, and NUSWIDEOBJ, as shown in Fig. 5. From the Fig. 5, we can see that most of the datasets tend to converge in 12 iterations.

Influence of Prior Anchor Graph Regularization

In this section, in order to verify the effectiveness of the proposed prior anchor graph regularization (PAGG), we destroy the PAGG function by setting $\beta = 0$, which is actually the comparison method CAG. Then, we verify the effectiveness of PAGG from two aspects, namely clustering performance and anchor distribution. From the perspective of clustering performance, our proposed SMGC is superior to CAG.

Furthermore, we take UCI-DIGIT dataset as an example to draw the distribution of anchors, which the number of samples, views and clusters of UCI-DIGIT are 2000, 3 and 10, respectively (Tang et al. 2021). Because the number of clusters in UCI-DIGIT dataset is relatively small, it is easy to display. From Fig. 6, it is not difficult to see that the anchors selected by SMGC are uniform distributed in each cluster. However, CAG method select anchors that are distributed inhomogeneous in each cluster, as shown in Fig. 2 (a). This is because our proposed SMGC makes use of the prior anchor labels (PAL) to make the anchors with discriminative cluster structure and uniform distributed property.

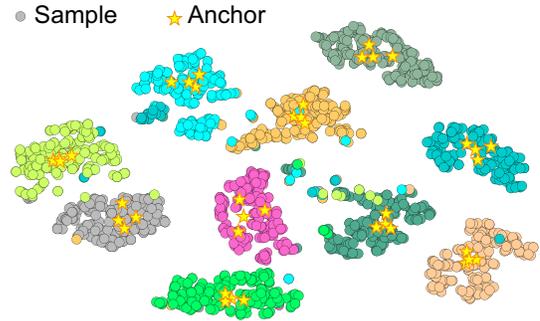


Figure 6: Visualization of anchors on UCI-DIGIT dataset.

Computational Time Comparison

For fairness, we compare the computational time on six benchmark datasets in detail. It is worth noting that unrecorded datasets cannot be tested because of insufficient memory. Moreover, the comparison methods are not recorded due to the inability to test too many results. As can be seen from Fig. 7, the computation time of our proposed SMGC is significantly lower than MVC methods based on subspace learning. Moreover, it is equal to the MVC methods based on anchor, which proves that the PAGG does not increase the computation time too much.

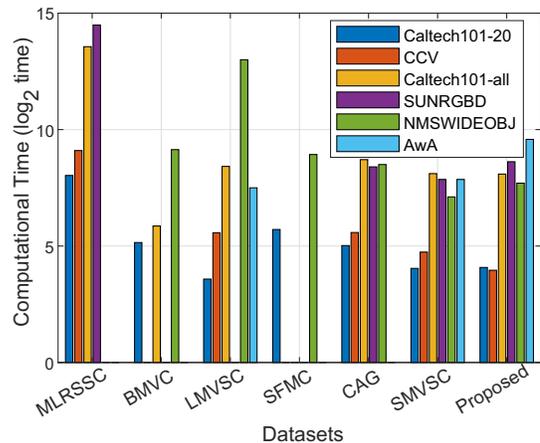


Figure 7: Computational time of different methods.

Conclusion

In this paper, we propose a novel MKC method, *i.e.*, SMGC, which innovatively propose PAGG using the PAL to constrain anchor selection. By doing this, the dynamically selected anchors are uniform distributed in different clusters and have a discriminative cluster structure. Next, we learn the similarity relationship between optimal anchors and original data points, and then obtain a high quality bipartite graph. Moreover, the k -means is applied to bipartite graph to obtain the clustering result. Extensive experiments prove the effectiveness and efficiency of the proposed SMGC. In the future, we plan to further study how to improve the distribution and representativeness of anchors.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 62106209), the Project of SongShan Laboratory (Grant no. 202215), the Project of Key Laboratory of System Control and Information Processing (Grant no. Scip202210), and the Sichuan Science and Technology Program (Grant no. 2023NSFSC0506).

References

- Brbić, M.; and Kopriva, I. 2018. Multi-view low-rank sparse subspace clustering. *Pattern Recognition*, 73: 247–258.
- Du, G.; Zhou, L.; Yang, Y.; Lü, K.; and Wang, L. 2021. Deep multiple auto-encoder-based multi-view clustering. *Data Science and Engineering*, 6(3): 323–338.
- Kang, Z.; Guo, Z.; Huang, S.; Wang, S.; Chen, W.; Su, Y.; and Xu, Z. 2019. Multiple partitions aligned clustering. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2701–2707.
- Kang, Z.; Zhao, X.; Peng, C.; Zhu, H.; Zhou, J. T.; Peng, X.; Chen, W.; and Xu, Z. 2020. Partition level multiview subspace clustering. *Neural Networks*, 122: 279–288.
- Lemmetty, M.; Keskinen, J.; and Marjamäki, M. 2005. The ELPI response and data reduction II: Properties of kernels and data inversion. *Aerosol science and technology*, 39(7): 583–595.
- Li, X.; Zhang, H.; Wang, R.; and Nie, F. 2020. Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1): 330–344.
- Liu, S.; Wang, S.; Zhang, P.; Xu, K.; Liu, X.; Zhang, C.; and Gao, F. 2022. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7576–7584.
- Liu, X.; Liu, L.; Liao, Q.; Wang, S.; Zhang, Y.; Tu, W.; Tang, C.; Liu, J.; and Zhu, E. 2021. One pass late fusion multi-view clustering. In *International Conference on Machine Learning*, 6850–6859. PMLR.
- Lu, X.; Feng, S.; Lyu, G.; Jin, Y.; and Lang, C. 2022. Distance-Preserving Embedding Adaptive Bipartite Graph Multi-View Learning with Application to Multi-Label Classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*.
- Ng, A. 2000. CS229 Lecture notes. *CS229 Lecture notes*, 1(1): 1–3.
- Ou, Q.; Wang, S.; Zhou, S.; Li, M.; Guo, X.; and Zhu, E. 2020. Anchor-based multiview subspace clustering with diversity regularization. *IEEE MultiMedia*, 27(4): 91–101.
- Qiang, Q.; Zhang, B.; Wang, F.; and Nie, F. 2021. Fast multi-view discrete clustering with anchor graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9360–9367.
- Sun, M.; Zhang, P.; Wang, S.; Zhou, S.; Tu, W.; Liu, X.; Zhu, E.; and Wang, C. 2021. Scalable multi-view subspace clustering with unified anchors. In *Proceedings of the 29th ACM International Conference on Multimedia*, 3528–3536.
- Tang, Y.; Xie, Y.; Zhang, C.; and Zhang, W. 2021. Constrained Tensor Representation Learning for Multi-View Semi-Supervised Subspace Clustering. *IEEE Transactions on Multimedia*.
- Wang, R.; Lu, J.; Lu, Y.; Nie, F.; and Li, X. 2022. Discrete and Parameter-Free Multiple Kernel k-Means. *IEEE Transactions on Image Processing*, 31: 2796–2808.
- Wang, S.; Liu, X.; Zhu, X.; Zhang, P.; Zhang, Y.; Gao, F.; and Zhu, E. 2021. Fast parameter-free multi-view subspace clustering with consensus anchor guidance. *IEEE Transactions on Image Processing*, 31: 556–568.
- Wang, X.; Lei, Z.; Guo, X.; Zhang, C.; Shi, H.; and Li, S. Z. 2019. Multi-view subspace clustering with intactness-aware similarity. *Pattern Recognition*, 88: 50–63.
- Wang, Y.; and Wu, L. 2018. Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering. *Neural Networks*, 103: 1–8.
- Wu, D.; Dong, X.; Nie, F.; Wang, R.; and Li, X. 2022. An attention-based framework for multi-view clustering on Grassmann manifold. *Pattern Recognition*, 128: 108610.
- Yang, B.; Zhang, X.; Chen, B.; Nie, F.; Lin, Z.; and Nan, Z. 2022a. Efficient correntropy-based multi-view clustering with anchor graph embedding. *Neural Networks*, 146: 290–302.
- Yang, H.; Gao, Q.; Xia, W.; Yang, M.; and Gao, X. 2022b. Multi-view Spectral Clustering with Bipartite Graph. *IEEE Transactions on Image Processing*.
- You, J.; Hou, Y.; Ren, Z.; You, X.; Dai, J.; and Yao, Y. 2022. Cluster Center Consistency Guided Sampling Learning for Multiple Kernel Clustering. *Information Sciences*.
- Zhan, K.; Nie, F.; Wang, J.; and Yang, Y. 2018. Multiview consensus graph clustering. *IEEE Transactions on Image Processing*, 28(3): 1261–1270.
- Zhang, P.; Liu, X.; Xiong, J.; Zhou, S.; Zhao, W.; Zhu, E.; and Cai, Z. 2020. Consensus One-step Multi-view Subspace Clustering. *IEEE Transactions on Knowledge & Data Engineering*, (01): 1–1.
- Zhang, Z.; Liu, L.; Shen, F.; Shen, H. T.; and Shao, L. 2018. Binary multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1774–1782.
- Zhou, S.; Ou, Q.; Liu, X.; Wang, S.; Liu, L.; Wang, S.; Zhu, E.; Yin, J.; and Xu, X. 2021. Multiple kernel clustering with compressed subspace alignment. *IEEE Transactions on Neural Networks and Learning Systems*.