INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 1, Month: February, Year: 2026
Article Number: 7103, https://doi.org/10.15837/ijccc.2026.1.7103

CCC Publications

Multi-View Clustering Method Based on Bipartite Graph Matrix Consistency

Dan Luo, Ting Chen

Dan Luo
College of Computer and Software
Chengdu Jincheng College, China
No. 1, Xiyuan Avenue, Chengdu City, Sichuan Province, China
luodan_1985@163.com

Ting Chen*
School of Intelligent Science and Technology
TianFu College of Southwestern University of Finance and Economics, China
No. 2, YuanXing West Street, Mianyang City, Sichuan Province, China
*Corresponding author: chengt@tfswufe.edu.cn

Abstract

To improve the performance and adaptability of multi-view clustering and address issues such as the neglect of view consistency information in graph construction, sensitivity to initial values, and the inability to adaptively learn view weights in existing algorithms, this paper proposes a Multi-View Clustering method based on Bipartite Graph Matrix Consistency (BGMC). The method learns consistency information represented by consistent anchor points across multiple views, jointly optimizes the similarity bipartite graphs of each view, and uses an alternating iterative strategy to solve for the optimal bipartite graph matrix. The model integrates view weights, a unified matrix, anchor matrices, and similarity matrices into a single optimization framework and introduces an anchor point mechanism to reduce computational complexity.Experiments on five real-world datasets including 3sources and YouTube Faces show that BGMC achieves an ACC 3-8 percent higher than the optimal method, an NMI 5-10 percent higher, and a convergence speed improved by over 20 percent.

Keywords: Consistent Information, Multi-view Clustering, Bipartite Graph Matrix, Self-adaptation.

1 Introduction

With the rapid development of internet and communication technologies, data from the real world can be extracted from multiple data sources. Multi-view clustering, which aims to fully utilize the information contained in multiple views to improve clustering performance, has become a research hotspot[16]. The objective of multi-view clustering is to integrate feature information from multiple views. Through joint learning, similar samples are grouped into the same cluster, and dissimilar ones into different clusters[15]-[8]. Consistent partitioning across views is required to ensure clustering accuracy and stability.

However, the current algorithms still suffer from several limitations. During the graph construction phase, they excessively depend on the conditional independence assumption of multi-view data, thus overlooking the discrepancies among different views and the consistency information across views. In the graph fusion process, both the pre-defined anchor point set and the learned view graph matrices remain static, making them highly sensitive to initial values and prone to converging to local optima. Additionally, most algorithms fail to adaptively learn the weight of each view without introducing extra hyperparameters.

To tackle the aforementioned challenges, this paper presents a Multi-view Clustering approach based on Bipartite Graph Matrix Consistency (MVC-BGMC). The method learns unified anchor points that encapsulate consistent information across multi-views to reduce the influence of biases inherent in individual views. It dynamically calculates view weights by measuring the discrepancy between each view and the unified graph, thereby amplifying the contribution of critical views. Through joint learning of similarity bipartite graphs for all views, an optimal bipartite graph is derived. By integrating view weights, a unified matrix, an anchor matrix, and a similarity matrix into a single optimization framework, the model significantly enhances algorithmic stability. Additionally, an anchor mechanism is introduced during bipartite graph construction, which boosts computational efficiency while maintaining clustering performance.

2 Related Works

2.1 Graph-based Multi-view Clustering Algorithms

Graph-based multi-view clustering characterizes the relationships between samples by constructing similarity graphs and improves clustering performance through graph learning techniques. Depending on diverse graph construction approaches, multi-view clustering algorithms can be categorized into two major types: those based on similarity graphs and those relying on bipartite graphs.[1].

Similarity graph-based multi-view clustering algorithms solve for an $n \times n$-dimensional similarity matrix in the initial graph construction stage to represent pairwise similarities between data points. After learning the initial graphs for different views, the algorithm proposed by Kim et al.[11] introduces rank constraints for further optimization, and the integrated global graph can reflect the exact number of cluster connection components. Biskel[3] not only focused on intra-view relationships but also considered inter-view similarities. Initially, they derived a similarity matrix for each individual view, and then integrated the acquired information into a clustering indicator matrix. They put forward two parameter-free strategies—the self-weighted approach and the adaptive weighting approach—to evaluate the weights of different views, thus eliminating the need for excessive parameter tuning. The algorithm proposed by Wang[17] achieves mutual constraint and update of the required matrices and clustering results within one framework. Under the constraint of the Laplacian matrix's rank on the graph, the quantity of connected components in the combined graph is equivalent to the specified number of clustering categories.

Khan et al.[10] proposed a scalable graph learning framework where bipartite graphs, clustering indicator matrices, and view weights are mutually learned with adaptive supervision. Connectivity constraints are employed to ensure that components directly represent clusters, and a connection between the proposed algorithm and K-means clustering is further established. Arthur et al.[1] proposed a multi-view clustering method via effective anchor representation learning, which mainly consists of three stages: anchor selection, consensus representation learning, and K-means clustering. This model circumvents the need to compute pairwise similarity between samples. Mi et al.[13] pointed out that the selected anchors fail to fully characterize the intrinsic data structure and first proposed to learn anchors within an embedded subspace. To ensure structural balance, a novel balancing strategy was devised to regulate the learned labels.

Graph-based multi-view clustering algorithms have strong universality and can handle convex data and data spaces of arbitrary shapes. However, it can be found that such algorithms rely on the number of initial clustering centers and the initialization of the initial graph, and the quality of the initial graph is difficult to guarantee, requiring parameter setting based on experience.

2.2 Subspace-based Multi-view Clustering Algorithms

Subspace-based multi-view clustering captures the latent manifold topology of multi-view data by learning shared low-dimensional subspaces, thereby achieving sample clustering. It mainly includes three steps: First, the self-expressiveness of the data is used to learn the subspace representations of different views, which are fused to obtain a unified representation. Further, a certain clustering method is applied to the unified representation to obtain clustering results.

Kevin et al.[9] first attempted to address large-scale data problems using subspace clustering. They first constructed a smaller graph between the original data and generated anchor points, and designed a new fusion mechanism to merge these graphs, which can significantly accelerate feature decomposition. The proposed algorithm has a linear time complexity. Winn et al.[18] fully considered the physical meaning of singular values and proposed using low-rank tensors to represent the high-order relationships between views. Chen et al.[5] introduced a diverse representation module into the network to enhance the discriminability of the learned representation matrix. Based on the deep subspace clustering network, they integrated block-diagonal and multivariate representations to develop a block-diagonal and multivariate-representation-integrated multi-view subspace clustering architecture.

The starting point of multi-view clustering algorithms based on subspace representation learning is to address the challenge of clustering high-dimensional data. The clustering process fully considers the specific information of each view, and subspace representation can also reduce the impact of noise on clustering performance. However, such algorithms also have the problem that the generated low-dimensional representations are difficult to interpret.

3 Research Methods

This paper combines bipartite graph construction with consistent graph learning. The bipartite graph leverages an anchor mechanism to effectively capture the local structural features of views, reducing computational complexity while enhancing noise robustness. A unified matrix enforces cross-view alignment of anchors to capture global structures. Their integration achieves a unified modeling of local features and global consistency, thus significantly boosting multi-view clustering performance. An adaptive weighting mechanism is introduced to adjust view importance, suppress noisy views, and highlight the contribution of critical views.

3.1 Notation

For a multi-view dataset with m views, the specific notation definitions are shown in Table 1.

3.2 Unified Graph Learning

This paper learns a bipartite graph based on the similarity between data points and their corresponding adjacent anchor points.When $\{A^v\}_{v=1}^m$ is fixed, the graphs of all views can be learned in the following form:

$$\min_{\{S^v\}_{v=1}^m} \sum_{v=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{t}\|x_i^v - a_j^v\|_2^2 s_{ij}^v + \alpha\sum_{v=1}^{m}\|S^v\|_F^2 \quad \text{s.t. } \forall v, s_{ij}^v \geq 0, \mathbf{1}^T s_i^v = 1 \tag{1}$$

In the above equation, the parameter α is used to control the connection sparsity between data points and multiple anchor points. If $\alpha = 0$, problem (1) has a solution, i.e., $s_{ij}^v = 1$, which means that only its nearest anchor point a_j^v can be connected to x_i^v. This is called hard partitioning. If α is large enough, connections from all t anchor points $\left\{a_j^v\right\}_{j=1}^t$ to x_i^v can be established with the same probability of $1/t$. The normalization $\mathbf{1}^T s_i^v = 1$ can be regarded as a sparsity constraint on S^v.

Table 1: Notation and Definitions

Notation	Definitions and Explanations	Dimension / Type
X^v	The data matrix of the $v-th$ view	$\mathbb{R}^{d_v \times n}$
x_i^v	The $i-th$ sample (column vector) in the $v-th$ view	$\mathbb{R}^{d_v \times 1}$
A^v	The anchor matrix of the $v-th$ view, which contains t anchors	$\mathbb{R}^{d_v \times t}$
a_j^v	The $j-th$ anchor in the $v-th$ view	$\mathbb{R}^{d_v \times 1}$
S^v	The similarity matrix between the data and anchors in the $v-th$ view	$\mathbb{R}^{n \times t}$
s_ij^v	The connection probability between the sample x_i^v and the anchor a_j^v	$\mathbb{R}^{d_v \times 1}$
U	Unified bipartite graph matrix, integrating multi-view similarity	$\mathbb{R}^{n \times t}$
δ_v	The weight of the $v-th$ view	Scalar
F	Feature matrix	$\mathbb{R}^{(n+t) \times c}$
D_U	The degree matrix of the bipartite graph	$\mathbb{R}^{(n+t) \times (n+t)}$
Z	The weighted adjacency matrix of the bipartite graph	$\mathbb{R}^{(n+t) \times (n+t)}$
L_U	Normalized Laplacian matrix	$\mathbb{R}^{(n+t) \times (n+t)}$
$\lVert \boldsymbol{X} \rVert_F$	Frobenius norm	Scalar
c	The number of target clusters	Scalar
t	The total number of anchors	Scalar

Each graph is only interconnected through the anchor sets. Therefore, views are learned independently by constructing a similarity matrix for each view when the anchor sets are fixed. Then, a unified bipartite graph matrix is generated and used to adaptively update $\{A^v\}_{v=1}^m$ until convergence.

A unified bipartite graph can be constructed by an integrated matrix $U \in R^{n \times t}$ using the collection of $\{S^v\}_{v=1}^m$. Hence, the following issue requires systematic exploration:

$$\min_U \sum_{v=1}^m \|U - S^v\|_F^2 \delta_v \qquad \text{s.t.} \forall i, u_{ij} \geq 0, \mathbf{1}^T u_i = 1 \tag{2}$$

Where δ_v denotes the weight of the $v-th$ view, $\delta = \{\delta_1, ..., \delta_m\}$, and their sum is 1. Although introducing view weights can improve clustering performance, Equation (2) yields a trivial solution, where the weight of the best view is assigned 1 and the weights of other views are 0. Therefore, the algorithm introduces a new regularization term $\alpha \ln \alpha$, which not only avoids the trivial solution but also generates a more stable optimal bipartite graph. Thus, Equation (2) can be rewritten as follows:

$$\min_U \sum_{v=1}^m \delta_v \|U - S^v\|_F^2 + \gamma \sum_{v=1}^m \delta_v \ln \delta_v \tag{3}$$

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

$$\begin{aligned} \min_{\{S^v\}_{v=1}^m, U} \sum_{v=1}^m \sum_{i=1}^n \sum_{j=1}^t \|x_i^v - a_j^v\|_2^2 s_{ij}^v + \alpha \sum_{v=1}^m \|S^v\|_F^2 + \sum_{v=1}^m \|U - S^v\|_F \delta_v \\ \text{s.t.} \forall v, i, s_{ij}^v \geq 0, \mathbf{1}^T s_i^v = 1, u_{ij} \geq 0, \mathbf{1}^T u_i = 1 \end{aligned} \tag{4}$$

When $\{A^v\}_{v=1}^m$ is fixed, the matrices $\{S^v\}_{v=1}^m$ and U can be learned jointly.

3.3 Common Anchor Learning

While updating the unified matrix U, common anchors can be explored and repositioned across all views. For the $j-th$ sub-cluster of the data in the $v-th$ view, its anchor a_j^v is calculated by the average value of all associated data points connected to it.

$$a_j^v = \frac{\sum_{i=1}^n u_{ij} x_i^v}{\sum_{i=1}^n u_{ij}} \tag{5}$$

Where $a_j^v \in R^{d_v \times 1}$, $j = 1, ..., t$. Then the anchor matrices $\{A^v\}_{v=1}^m$ are updated. Finally, Equation (5) is combined with Problem (4) to jointly learn the matrices $\{S^v\}_{v=1}^m$, U, and $\{A^v\}_{v=1}^m$.

$$\begin{aligned} \min_{\{S^v\}_{v=1}^m, \{A^v\}_{v=1}^m, U} \sum_{v=1}^m \sum_{i=1}^n \sum_{j=1}^t \|x_i^v - a_j^v\|_2^2 s_{ij}^v + \alpha \sum_{v=1}^m \|S^v\|_F^2 + \sum_{v=1}^m \|U - S^v\|_F^2 \delta_v \\ \text{s.t.} \forall v, i, s_{ij}^v \geq 0, \mathbf{1}^T s_i^v = 1, u_{ij} \geq 0, \mathbf{1}^T u_i = 1 \end{aligned} \tag{6}$$

3.4 Optimal Bipartite Graph Learning

The edge weights of the bipartite graph are encoded by the matrix $U \in R^{n \times t}$, wherein each element u_{ij} is the edge weight connecting x_i and the corresponding a_j across all views. The block structures of the weighted adjacency matrix $Z \in R^{(n+t)\times(n+t)}$ and the degree matrix D_U are as follows:

$$Z = \begin{bmatrix} 0 & U \\ U^T & 0 \end{bmatrix}, D_U = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \tag{7}$$

Where $D_1 \in R^{n \times n}$, the $i - th$ vector of D_1 is $d_i^1 = \sum_{j=1}^t u_{ij}$; $D_2 \in R^{t \times t}$, the $j - th$ vector of D_2 is $d_j^2 = \sum_{i=1}^n u_{ij}$. Therefore, the normalized Laplacian matrix is:

$$L_U = \mathbf{I} - (D_U)^{-1/2} Z (D_U)^{-1/2} \tag{8}$$

Once the bipartite graph contains exactly c connected components, the neighborhood anchor allocation across all views achieves optimality for each data point. This can be realized by enforcing a rank constraint on the Laplacian matrix L_U associated with the bipartite graph Z of U. Reference[21] demonstrates that the multiplicity of the eigenvalue 0 for the normalized Laplacian matrix L_U is exactly c, which equals the number of connected components in the bipartite graph associated with U. If $rank\,(L_U) = (n+t) - c$, the n data points and t anchors can be partitioned into c clusters based on Z associated with U. Therefore, the final sub-clusters can be generated without performing additional clustering methods. The learning of the optimal bipartite graph is solved by Problem (9):

$$\begin{aligned} \min_{\{S^v\}_{v=1}^m, \{A^v\}_{v=1}^m, U} \sum_{v=1}^m \sum_{i=1}^n \sum_{j=1}^t \|x_i^v - a_j^v\|_2^2 s_{ij}^v + \alpha \sum_{v=1}^m \|S^v\|_F^2 + \sum_{v=1}^m \|U - S^v\|_F^2 \delta_v \\ \text{s.t.} \forall v, i, s_{ij}^v \geq 0, \mathbf{1}^T s_i^v = 1, u_{ij} \geq 0, \mathbf{1}^T u_i = 1, rank\,(L_U) = (n+t) - c \end{aligned} \tag{9}$$

To relax the constraint on the rank, this paper introduces the c smallest eigenvalues of L_U, denoted as $\{\eta_q\,(L_U)\}_{q=1}^c$, where $\eta_q\,(L_U) \geq 0$. Given that L_U is positive semi-definite, the rank constraint can be enforced via setting $\sum_{q=1}^c \eta_q\,(L_U) = 0$. This problem is reformulated as an optimization problem. The objective function can be obtained by substituting Problem (10) into Problem (9):

$$\sum_{q=1}^c \eta(L_U) = \min_{F \in \mathbb{R}^{(n+t)\times c}, F^T F = \mathbf{I}} \mathrm{Tr}(F^T L_U F) \tag{10}$$

In equation (10), F denotes a positive definite matrix.

$$\begin{aligned} \min_{\{S^v\}_{v=1}^m, \{A^v\}_{v=1}^m, F} \sum_{v=1}^m \sum_{i=1}^n \sum_{j=1}^t \|x_i^v - a_j^v\|_2^2 s_{ij}^v + \alpha \sum_{v=1}^m \|S^v\|_F^2 + \sum_{v=1}^m \|U - S^v\|_F^2 \delta_v + \beta \mathrm{Tr}(F^T L_U F) \\ \text{s.t.} \forall v, i, s_{ij}^v \geq 0, \mathbf{1}^T s_i^v = 1, u_{ij} \geq 0, \mathbf{1}^T u_i = 1, F^T F = \mathbf{I} \end{aligned} \tag{11}$$

As the parameter β is sufficiently large, the optimal solution U derived from Problem (11) can satisfy the condition $\sum_{q=1}^c \eta_q\,(L_U) = 0$. Additionally, β serves as a control parameter for the number of connected components in the bipartite graph, designated as γ. In each iteration, β increases when $\gamma < c$ and decreases when $\gamma > c$. Therefore, the resulting bipartite graph matrix Z yields exactly c connected components, grouping the n data points and t anchors into c clusters.

3.5 Optimization Strategy

The variables to be optimized in this paper are $\{S_v\}_{v=1}^m$, $\{\delta_v\}_{v=1}^m$, U, F, and $\{A_v\}_{v=1}^m$. The optimization method adopted here is to fix all other variables and update only one variable at a time.

3.5.1 Update S_v

When fixing δ_v, U, F and A_v, the last term becomes a constant, the problem becomes:

$$\min_{\{S^v\}_{v=1}^m} \sum_{v=1}^m \sum_{i=1}^n \sum_{j=1}^t \|x_i^v - a_j^v\|_2^2 s_{ij}^v + \alpha \sum_{v=1}^m \|S^v\|_F^2 + \sum_{v=1}^m \|U - S^v\|_F^2 \delta_v \quad \text{s.t.} \forall v, i, s_{ij}^v \geq 0, \mathbf{1}^T s_i^v = 1 \tag{12}$$

The update of S_v is independent across all views, and the update of each vector s_i^v is independent. Only consider the k nearest anchors for each data point x_i^v. The final optimization solution for s_{ij}^v is:

$$s_{ij}^v = \begin{cases} \frac{\theta_{i,k+1} - \theta_{ij} + 2\delta_v (u_{ij} - u_{i,k+1})}{k\theta_{i,k+1} - \sum_{a=1}^k \theta_{ia} - 2k\delta_v u_{i,k+1} + 2\sum_{a=1}^k \delta_v u_{ia}} & j \leq k \\ 0 & j > k \end{cases} \tag{13}$$

3.5.2 Update U

When fixing S_v, δ_v, F and A_v, Problem (11) can be transformed into:

$$\min_U \sum_{v=1}^m \|U - S^v\|_F^2 \delta_v + \beta \mathrm{Tr}(F^T L_U F) \quad \text{s.t.} \quad \forall i, u_{ij} \geq 0, \mathbf{1}^T u_i = 1 \tag{14}$$

All L_U, D_U, and Z depend on U, and each row update u_i in U is independent. Let φ and ϕ be the Lagrangian multipliers for the two constraints, so the Lagrangian function of Problem (14) can be obtained as:

$$L(u_i, \phi, \varphi) = \sum_{v=1}^m \|u_i - s_i^v\|_2^2 \delta_v + \beta \mu_i^T u_i - \phi(\mathbf{1}^T u_i - 1) - \varphi^T u_i \tag{15}$$

Then, take the derivative of L with respect to u_i, set it to zero, β can be adaptively determined, so β is treated as a known parameter. Let $a = 2\sum_{v=1}^m \delta_v$ and $p_i = 2\sum_{v=1}^m s_i^v \delta_v - \beta\mu_i$ be constants. According to the KKT conditions, we have:

$$u_i^* a - p_i - \phi^* \mathbf{1} - \varphi^* = \mathbf{0} \tag{16}$$

According to the constraint, we can obtain:

$$\phi^* = \frac{a - \mathbf{1}^T p_i - \mathbf{1}^T \varphi^*}{t} \tag{17}$$

Substituting ϕ^* into equation (16), we can get u_i^* for each row of U .

$$u_i^* = \frac{p_i}{a} + \frac{\mathbf{1}}{t} - \frac{\mathbf{1}^T p_i \mathbf{1}}{at} - \frac{\mathbf{1}^T \phi^* \mathbf{1}}{at} + \frac{\phi^*}{a} \tag{18}$$

3.5.3 Update F

When fixing S_v, δ_v, U and A_v, as a block matrix, F can be rewritten as:

$$F = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{19}$$

Where $F_1 \in R^{n \times c}$, $F_2 \in R^{t \times c}$, F can be updated by equation (19).

$$\max_{F_1^T F_1 F_2^T F_2 = \mathbf{I}} \mathrm{Tr}(F_1^T (D_1)^{-1/2} U (D_2)^{-1/2} F_2) \tag{20}$$

In equation (20), $B = (D_1)^{-1/2} U (D_2)^{-1/2}$, with $F_1 \in R^{n\times c}$, $B \in R^{n\times t}$ and $F_2 \in R^{t\times c}$ given. The optimal solution to the problem is:

$$\max_{F_1^T F_1 F_2^T F_2 = \mathbf{I}} \mathrm{Tr}(F_1^T B F_2) \tag{21}$$

Where $F_1 = \frac{\sqrt{2}}{2} B_1$ and $F_2 = \frac{\sqrt{2}}{2} B_2$, B_1 is the first c left singular vectors of B, and B_2 is the first c right singular vectors of B. The optimal F is composed of the optimal F_1 and F_2.

3.5.4 Update A_v

When fixing S_v, δ_v, U, and F, each a_j^v can be updated by Equation (4).

3.5.5 Update δ_v

When fixing S_v, U, F and A_v,let $h_\nu = \sum_{\nu=1}^m \|U - S^\nu\|_F^2$,the optimization of δ_v can be simplified to:

$$\sum_{v=1}^m \delta_v h_v + \gamma \sum_{v=1}^m \delta_v \ln \delta_v \tag{22}$$

The Lagrangian function of the above formula is:

$$\mathcal{L}(\delta_v, \eta) = \sum_{v=1}^m \delta_v h_v + \gamma \sum_{v=1}^m \delta_v \ln \delta_v - \eta \left(\sum_{v=1}^m \delta_v - 1 \right) \tag{23}$$

Where η is the Lagrangian multiplier, taking the partial derivative of $\mathcal{L}$ with respect to δ_v and setting it to zero, we have:

$$\frac{\partial \mathcal{L}}{\partial \delta_v} = h_v + \gamma(\ln \delta_v + 1) - 1 = 0 \tag{24}$$

By adding the constraint $\sum_{\nu=1}^m \delta_\nu = 1$ to the obtained solution, we can get:

$$\delta_v = \frac{\exp(-h_v/\gamma)}{\sum_{v=1}^m \exp(-h_v/\gamma)} \tag{25}$$

With all update processes completed, the steps of the proposed algorithm are as follows.

- Input: Datasets of m views $X^1, ..., X^m$, where $X^v \in R^{d_v \times n}$, the number of anchor points t, the number of clusters c, the number of anchor neighborhoods k, and the initial parameter β.
- Output: Clustering results.
- 1: Initialize the anchor point matrix A^v and the similarity matrix S^v.
- 2: Initialize the bipartite graph matrix U using Equation (2).
- 3: Set the weight of each view as $\delta_v = 1/m$.
- 4: Initialize F with the eigenvectors of the Laplacian matrix.
- 5: while not converged do
- 6: Update S^v through Equation (13).
- 7: Update U through Equation (18).
- 8: Update F through Equation (21).
- 9: Update A^v through Equation (4).
- 10: Update δ_v through Equation (25).
- 11: end while.

3.6 Complexity and Convergence Analysis

3.6.1 Complexity Analysis

The computational complexity of the BGMC approach is primarily composed of six components, which respectively correspond to the initialization and update processes of variables.

The complexity of updating $\{S^v\}_{v=1}^m$ is $O(mnt)$, with m being the count of views, n is the number of data objects, t is the number of anchor points with $c \leq t << n$, and c is the desired number of clusters.

The computational complexity of updating the weights of all views δ is $O(mnt)$.

The computational complexity of updating the unified graph matrix U is $O(cn)$.

The computational complexity of learning F is $O(cnt)$. Therefore, the computational complexity of this sub-iterative process is $O((2mt + c + ct)\, n\zeta_1)$, where ζ_1 is the number of iterations.

The computational complexity of updating the anchor points A is $O(mntd)$, where $d = max(d^1, ..., d^m)$.

The anchor points $\{A_v\}_{v=1}^m$ are initialized with a computational complexity of $O(ndt)$, and the initialization of $\{S^v\}_{v=1}^m$ has a computational complexity of $O(mntd)$.

Therefore, BGMC 's computational complexity amounts to $O(((2mt + c + ct)\, \zeta_1 + mtd)\, n\zeta_2 + ndt\,(m + 1))$, where ζ_2 represents the iteration count.

3.6.2 Convergence Analysis

Using the optimization strategy proposed earlier to solve problem (11), after alternately optimizing the variables, each corresponding subproblem is convex, and admits an optimal solution. Moreover, their convergence is demonstrated as follows.

During the update of $\{S^v\}_{v=1}^m$, the objective function in problem (13) exhibits convexity. Since its second derivative with respect to s_i^v equals 1, the optimization strategy adopted is monotonically decreasing.

When updating the weight δ, the objective function in problem (2) constitutes a linear convex problem, and the closed-form solution for δ is obtained by equation (25).

When updating U, $\hat{U}$ represents the updated value of U in the augmented Lagrangian iteration, with $\Gamma(U) = \beta Tr\left(F^T L_U F\right)$. Therefore, as the function error decreases, the following inequality can be deduced from problems (14):

$$\sum_{v=1}^{m} \frac{\|\widehat{U} - S^v\|_F^2}{2\|\widehat{U} - S^v\|_F} + \Gamma(\widehat{U}) \leq \sum_{v=1}^{m} \frac{\|\widehat{U} - S^v\|_F^2}{2\|\widehat{U} - S^v\|_F} + \Gamma(U) \tag{26}$$

According to the lemma in Reference[19], the convergence of Problem (14) can be obtained.

To update F, according to the lemma in Reference[19], the objective function of Problem (20) for updating F is updated through the singular value decomposition of B.

When updating $\{A_v\}_{v=1}^m$, the problem attains convergence once the associations between data points and anchor points remain unchanged.

4 Experiment

4.1 Dataset selection

To evaluate the effectiveness of BGMC, this paper carefully selects five representative multi-view datasets as experimental data, namely 3sources, bbcsport, Caltech-7, Mfeat, and Webkb, to comprehensively assess the performance of the proposed algorithm. The detailed information of these datasets is shown in Table 2.

In the table, v is the number of views, m is the number of data objects, c is the number of desired clusters, and dv represents the feature dimension in the $v - th$ view.

Table 2: Statistical Information of Experimental Datasets

Dataset	m	v	c	d^1	d^2	d^3	d^4	d^5	d^6
3sources	169	3	6	3560	3631	3068	–	–	–
bbcsport	116	4	5	1991	2063	2113	2158	–	–
Caltech - 7	1474	6	7	48	40	254	1984	512	928
STL10	13000	3	10	1024	512	2048	–	–	–
YouTube Faces	101499	5	31	64	512	64	647	838	–

4.2 Anchor Initialization Experiment

For the proposed BGMC method, the initialization of uniform anchor points plays a critical role, whereas K-means demonstrates notable sensitivity to the initial anchor positions. Considering the effectiveness of clustering and computational efficiency, four commonly used initialization methods are compared: k-Means[1], greedy k-means++[12], PCA-part[6], and Var-part[4]. All these initialization approaches process the concatenated features across all views.

Regarding the aforementioned initialization approaches, $d = d^1 + d^2 + \cdots + d^v$; l denotes the iteration count, t represents the number of centers, and s is the additional sampling size. In the experiment, one initialization method is chosen to initialize the anchor points of the BGMC method depicted in Step 1 in algorithm of Section 3.4.5. For each real-world dataset, $t = n/5$ and $k = 5$ are set empirically, and the initial value of the parameter β is set to 1. Its value is adaptively adjusted during the objective function optimization process for each dataset. Two commonly used clustering performance evaluation metrics are employed: Accuracy (ACC) and Normalized Mutual Information (NMI). For the randomized trials, each method is executed 5 times, and the mean values of the metrics are recorded.

Table 3 tabulates the clustering results of BGMC under two evaluation metrics using different initialization methods on five real-world datasets. Inspection of the table reveals that the PCA-part and Var-part initialization strategies exhibit comparatively similar performance, both yielding high-quality initial anchors that outperform k-means and greedy k-means++ in clustering accuracy. Computationally, the Var-part method demonstrates distinct advantages. While the PCA-part approach shows slightly better effectiveness in certain scenarios, the Var-part method offers better scalability to high-dimensional data with reduced computational complexity. For these reasons, this study selects the Var-part method as the default initialization strategy for the BGMC framework.

Table 3: Clustering Results of Different Initialization Methods on Five Real-World Datasets

Metric	Initialization method	3sources	bbcsport	Caltech - 7	STL10	YouTube Faces
ACC	K - Means	0.775	0.915	0.781	0.920	0.773
	Greedy k - means++	0.789	0.923	0.735	0.925	0.782
	PCA - Part	0.794	**0.929**	0.783	**0.933**	0.787
	Var - Part	**0.797**	0.921	**0.785**	0.932	**0.795**
NMI	K - Means	0.669	0.955	0.670	0.914	0.495
	Greedy k - means++	0.672	0.961	0.702	0.908	0.523
	PCA - Part	0.689	0.969	0.710	0.917	0.524
	Var - Part	0.705	0.960	0.697	0.910	0.540

4.3 Experimental Results

During the experiment, comparisons in terms of clustering will be made with the following algorithms.

(1) Single-view clustering algorithm: K-means is run on each view of the dataset, and the best clustering results among these views are recorded.

(2) Bipartite graph-based multi-view clustering algorithms: The LMVSC algorithm[7] constructs a smaller additional graph between the original data points and the generated anchor points, and

designs a new integration method to merge these graphs. The EMKMC algorithm[20] first constructs anchor point graphs for each view, then uses an improved K-means for integration, and designs two algorithms to solve the model.

(3) Graph-based multi-view clustering algorithm:The COMVSC[2] algorithm proposes a joint learning framework. By integrating partition-level information to reduce inter-data noise and effectively capture complementary information across views, it further improves the consistency and accuracy of clustering results.

(4) Deep Learning-Based Clustering Algorithms: The IBFDMVC[14] algorithm adopts a unique information-bottleneck fusion strategy, enabling it to handle the complexity of multi-view data. The SIB-SMC algorithm utilizes self-supervised learning and an information bottleneck strategy for multi-view subspace clustering.

According to the recommendations in the experimental analysis of existing papers, the parameters of the comparative methods were set. For BGMC, $t = n/5$ and $k = 5$ were set, respectively, with the initial parameter β ixed at 1. Its value was dynamically adjusted during the objective function optimization process for each dataset. Three widely adopted clustering evaluation metrics were utilized: ACC, NMI, and F- score. For the randomized trials, each method was executed five times, and the average metric values were reported. Table 4 presents the clustering results of the five methods across the five real-world datasets, with the optimal results highlighted in bold. Inspection of the table reveals that the BGMC method yields superior performance.

Table 4: Measurement Results of Clustering Five Real-World Datasets with Different Methods

Metric	method	3sources	bbcsport	Caltech - 7	STL10	YouTube Faces
ACC	K - Means	0.4961	0.2772	0.3726	0.5911	0.5038
	EMKMC	0.6568	0.6662	0.5160	0.6337	0.5727
	LMVSC	0.4994	0.5060	0.7266	0.7204	0.8965
	COMVSC	0.6864	0.6983	**0.8134**	0.7714	0.9450
	IBFDMVC	0.7960	0.7048	0.7368	0.7571	0.8965
	SIBSMC	0.7747	0.7156	0.7145	0.7245	0.8057
	BGMC	**0.8263**	**0.8056**	0.7859	**0.8950**	**0.9048**
NMI	K - Means	0.5356	0.4336	0.4710	0.6505	0.6108
	EMKMC	0.6737	0.7438	0.5440	0.6851	0.6183
	LMVSC	0.5754	0.5443	0.5193	0.7596	0.8443
	COMVSC	0.5301	0.5346	0.5311	0.7040	0.8925
	IBFDMVC	0.7465	0.7146	**0.7652**	0.8160	0.8323
	SIBSMC	0.7293	0.7257	0.7046	0.7267	0.8053
	BGMC	**0.7754**	**0.7532**	0.7082	**0.8363**	**0.8725**
F - score	K - Means	0.4556	0.4569	0.4654	0.6952	0.6320
	EMKMC	0.7160	0.7017	0.8130	0.7095	0.7125
	LMVSC	0.5207	0.6034	0.6947	0.7143	0.8540
	COMVSC	0.6788	0.5322	0.7728	0.6776	0.8934
	IBFDMVC	0.8054	0.7534	0.7854	0.8953	0.8834
	SIBSMC	0.7542	0.7835	0.7354	0.8457	0.8369
	BGMC	**0.8338**	**0.7935**	**0.8233**	**0.9091**	**0.9194**

5 Discussion

5.1 Analysis of Experimental Results

As observed from Table 4, in terms of ACC and NMI, the BGMC algorithm yields the optimal performance across all datasets, except for a near - optimal solution on the Caltech - 7 dataset. In terms of F - score, BGMC outperforms other baseline methods significantly. When evaluated by average values, BGMC also demonstrates superiority over all comparative methods. The BGMC algorithm

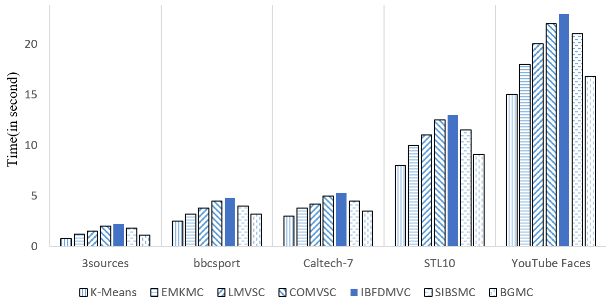

Figure 1: Comparison of Running Time

designates cluster centers as anchors for bipartite graph construction, and incorporates matrix - derived information for dynamic iterative updating. Compared with a complete graph, it can better capture fine - grained partition structure information across views. Moreover, by learning consistent anchors from individual graphs, a unified graph, and all views, BGMC enables the acquisition of a more robust unified graph representation.

5.2 Comparison of Running Time

Experiments were conducted for each clustering method in the same computational environment, and the running time of each method on each multi-view dataset was recorded in detail, as shown in Figure 1.

Experimental results indicate that when the dataset size is small, the running time of all algorithms is relatively short, whereas as the dataset scales up, the running time of these algorithms lengthens accordingly. Among multi - view approaches, BGMC exhibits remarkable time efficiency across all datasets. Notably, on medium - to large - scale datasets, its running time is significantly lower than that of traditional clustering algorithms (e.g., LMVSC). This efficiency enhancement is primarily attributed to the introduction of an anchor mechanism in the bipartite graph construction by BGMC, which successfully reduces the computational time complexity while safeguarding clustering performance. Furthermore, in comparison to K - means, which boasts shorter running time, BGMC achieves a more optimal balance between operational efficiency and clustering performance. This demonstrates its higher applicability in handling multi - view large - scale data.

5.3 Ablation experiment

To verify the effectiveness of the proposed algorithm, we conducted three ablation experiments by: R1) removing the anchor update mechanism; R2) fixing the view weights; and R3) removing spectral clustering, with each serving as a control variable. All experiments were performed under the same experimental environment and parameter settings. The results are shown in Table 5, where the optimal results are marked in bold.

Table 5: Measurement Results of Clustering Five Real-World Datasets with Different Methods

Metric	method	3sources	bbcsport	Caltech - 7	STL10	YouTube Faces
ACC	BGMC	**0.8263**	**0.7048**	**0.7359**	**0.7826**	**0.9048**
	R1	0.5063	0.6387	0.4946	0.6234	0.8025
	R2	0.5463	0.6254	0.5134	0.7571	0.8209
	R3	0.6063	0.5177	0.6134	0.5254	0.6383
F - score	BGMC	**0.8338**	**0.7935**	**0.8233**	**0.9091**	**0.9194**
	R1	0.7035	0.5973	0.7921	0.4354	0.8341
	R2	0.6279	0.7247	0.6526	0.7234	0.5033
	R3	0.6367	0.6635	0.6848	0.8345	0.5912

As shown in the ablation experiment results, when only fixed initial anchor points are used without

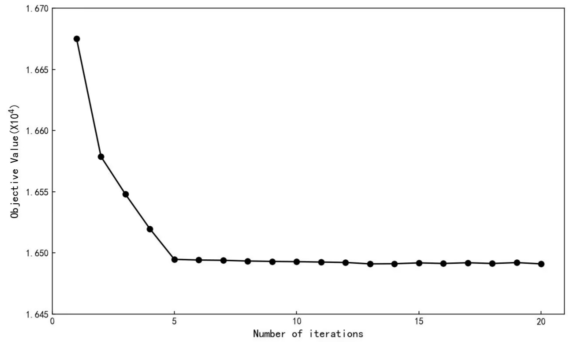

Figure 2: Convergence curves of BGMC on 3source

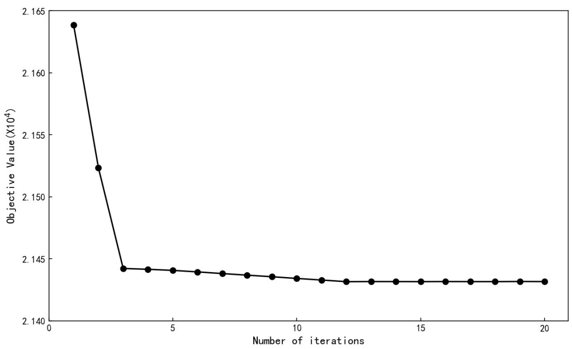

Figure 3: Convergence curves of BGMC on bbcsport

iteratively updating the anchor point matrix via a unified bipartite graph matrix, the ACC and F-score metrics across all datasets decrease. This indicates that the bipartite graph matrix is crucial for capturing view consistency across views. If the weights of each view are fixed statically at 1/m instead of being dynamically adjusted based on view quality, the NMI drops by 5-8 percent, demonstrating that dynamic weights can boost the contribution of key views. After removing spectral embedding, the clustering performance metrics on all datasets show a significant decline, further verifying the importance of fusing view information in improving clustering performance.

5.4 Convergence Analysis

To verify the effectiveness of the objective function optimization strategy proposed in this paper, the convergence curves of BGMC on five different datasets are plotted. For each subgraph, the x-axis represents the number of iterations, and the y-axis represents the objective function value. Convergence curves of BGMC on 3source as shown in Figure 2, convergence curves of BGMC on bbcsport as shown in Figure 3, convergence curves of BGMC on cal-7 as shown in Figure 4, convergence curves of BGMC on STL10 as shown in Figure 5, convergence curves of BGMC on YouTube Faces as shown in Figure 6. It can be seen from the figure that BGMC converges rapidly for all datasets. According to the convergence iterations of the comparative algorithms in existing papers, SIBSMC converges within 10 iterations, IBFDMVC converges after 15 iterations, and LMVSC converges after 20 iterations. In terms of the number of convergence iterations, the algorithm in this paper performs above the average level.

6 Conclusion

To fully exploit consistency information and enhance clustering performance and adaptability, this paper presents a novel multi-view clustering approach grounded in bipartite graph matrix consistency. By learning and leveraging the consistency information encapsulated by a sparse set of consistent anchor points, the proposed method jointly learns the similar bipartite graphs across multiple views

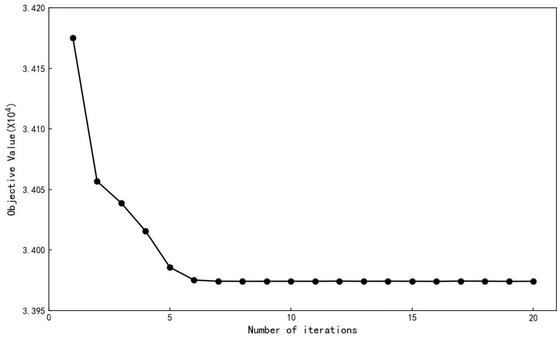

Figure 4: Convergence curves of BGMC on cal-7

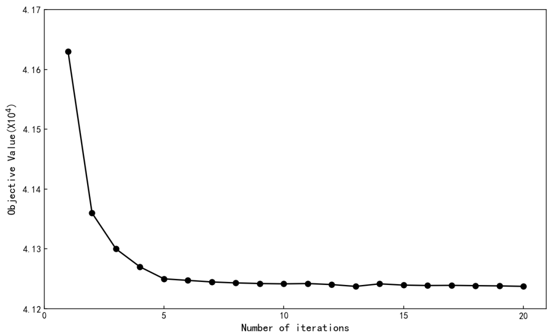

Figure 5: Convergence curves of BGMC on STL10

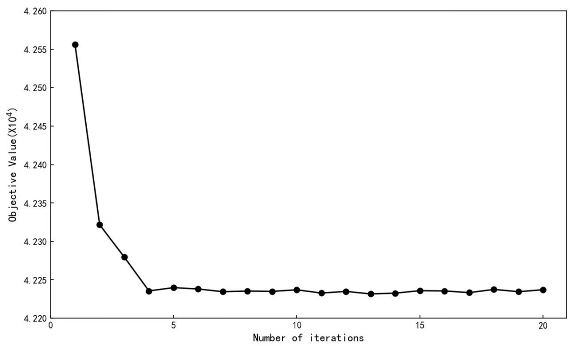

Figure 6: Convergence curves of BGMC on YouTube Faces

and employs an efficient alternating iterative optimization strategy to progressively resolve the variable optimization problem. Extensive experimental evaluations demonstrate that the method achieves superior performance in terms of clustering accuracy, convergence speed, and algorithmic efficiency. Nevertheless, it remains subject to certain limitations: the view weights exhibit insufficient robustness when confronted with views containing intense noise or substantial semantic disparities, and parameters such as the number of anchor points rely on empirical settings rather than fully automated determination. Future research may focus on directions including robust weight learning mechanisms, parameter-free anchor selection strategies, and cross-modal data extension, with the aim of further enhancing robustness, universality, and cross-scenario application capabilities.

References

[1] Arthur D, Vassilvitskii S. (2007). K-Means++: The advantages of careful seeding, *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027–1035,2007

[2] Åleskog C, Devagiri V M, Boeva V. (2022). A Graph - Based Multi - view Clustering Approach for Continuous Pattern Mining, *In: Studies in Big Data*, Cham: Springer, 2022.

[3] Biskel S. (2004). Multi-view clustering, *Proceedings of IEEE International Conference on Data Mining*, 2004.

[4] Cesar, L. M., Alonso, G. C. D., Octavio, V. L., Luis, V. C. J. (2024). MCClusteringSM: An approach for the Multicriteria Clustering Problem based on a Credibility Similarity Measure, *Computer Science and Information Systems*, 21(3), 1147–1177, 2024.

[5] Chen X, Sun W, Wang B,et al. (2019). Spectral clustering of customer transaction data with a two-level subspace weighting method, *IEEE Trans. Cybernetics*, 49(9), 3230–3241, 2019.

[6] Dong, Z., Chen, F., Zhang, Y. (2024). An Efficient Multiple Kernel Approach to Multi-class Classification, *Computer Science and Information Systems*, 21(1), 143–166, 2024.

[7] Galić, D., Stojanović, Z, Čajić, E. (2024). Application of Neural Networks and Machine Learning in Image Recognition, *Tehnicki vjesnik–Technical Gazette*, 31(1), 316–323, 2024.

[8] Karahan, M. (2024). Nonlinear Modelling and Robust Backstepping Control of a Quadcopter in Aggressive Maneuvering, *Studies in Informatics and Control*, 33(3), 29–38, 2024.

[9] Kevin, Fan. (2009). On a theorem of Weyl concerning eigenvalues of linear transformations, *Proceedings of the National Academy of Sciences of the United States of America*, 36(1), 31–35, 2009.

[10] Khan Ahmad Ghufran, Hu Jie, Li Tianrui, et al.(2022). Multiview data clustering via non-negative matrix factorization with manifold regularization, *International Journal of Machine Learning and Cybernetics*, 13(3), 677–689, 2022.

[11] Kim Seo Young, Won Lee J.(2007). Ensemble clustering method based on the resampling similarity measure for gene expression data, *Statistical Methods in Medical Research*, 16(6), 539–564, 2007.

[12] Koprinkova-Hristova, P., Georgiev, I., Raykovska, M.(2024). Echo State Network for Features Extraction and Segmentation of Tomography Images, *Computer Science and Information Systems*, 21(1), 379–393, 2024.

[13] Li Z F, Li H Y, Liu X W.(2018). Subspace clustering method based on classification sparse low-rank representation, *Journal of Systems Science and Mathematical Sciences*, 38(8), 852–865, 2018.

[14] Nie Feiping, Li Jing, Li Xuelong.(2017). Self-weighted multiview clustering with multiple graphs, *Proceedings of International Joint Conference on Artificial Intelligence*, 2564–2570, 2017.

[15] Rădulescu, I.-C.(2024). Robust Model Predictive Control for Systems Affected by Constant and Norm 2 Bounded Disturbance, *Studies in Informatics and Control*, 33(1), 99–106, 2024.

[16] Vesselenyi, Tiberiu and Dzitac, Ioan and Dzitac, Simona and Vaida, Victor .(2008). Surface Roughness Image Analysis using Quasi-Fractal Characteristics and Fuzzy Cluster-ing Methods, *International Journal of Computers Communications and Control*, 3(3), 304–316, 2008.

[17] Wang Lijuan, Ding Shifei, Ding Ling.(2020). Soft Subspace Clustering Algorithm Based on Transfer Learning, *Journal of Nanjing University (Natural Sciences)*, 56(4), 515–523, 2020.

[18] Winn J, Jojic N. Locus.(2005). Learning object classes with unsupervised segmentation, *Proceedings of the Tenth IEEE International Conference on Computer Vision*, 745–763, 2005.

[19] Yin, S., Wang, L., Teng, L..(2024). Threshold Segmentation Based on Information Fusion for Object Shadow Detection in Remote Sensing Images, *Computer Science and Information Systems*, 21(4), 1221–1241, 2024.

[20] Zhang, X.(2024). Visual Communication Design of Weak and Small Target Images Based on Image Processing Model and Data Fusion, *Tehnicki vjesnik–Technical Gazette*, 31(3), 815–823, 2024.

[21] Zou, Q., Liu, F., Liao, Y.(2024). Enhancing Architectural Image Processing: A Novel 2D to 3D Algorithm Using Improved Convolutional Neural Networks, *Computer Science and Information Systems*, 21(4), 1457–1481, 2024.

Cite this paper as:

Luo, D.; Chen, T. (2026). Multi-View Clustering Method Based on Bipartite Graph Matrix Consistency, *International Journal of Computers Communications & Control*, 21(1), 7103, 2026.
https://doi.org/10.15837/ijccc.2026.1.7103