# Auto-weighted Multi-view Subspace Clustering with Consistency Learning

*Ao Li[1], Yong Wang[1,2*], Lihao Yang[1], Shuai Meng[1]*

*[1] School of Computer and Information, Anhui Polytechnic University, China*
*[2] Ministry of Industry and Information Technology,*
*Nanjing University of Aeronautics and Astronautics (Key Laboratory of Safety-Critical Software), China*
*2220910101@stu.ahpu.edu.cn, yongwang@ahpu.edu.cn, lihaoyang020630@163.com, mengshuai2024@qq.com*

## Abstract

Multi-view subspace clustering (MSC) has received widespread attention due to its ability to efficiently exploit consensus and diversity information from multiple perspectives. However, existing methods focus more on inter-view diversity and ignore the consistent associations and higher-order features among views under different perspectives. To solve the above problems, an auto-weighted MSC with consistency learning (AWMSCC) is proposed. Specifically, this method first integrates the shared features of all coefficient matrices in a three-factor decomposition to construct a new shared consistency matrix. Then, by using the tensor low-rank constraint, the coefficient matrices and the shared consistency matrix are stacked into a third-order tensor to achieve effective propagation of inter-view consistency information. Finally, the appropriate weights are adaptively assigned to all matrix information to obtain a high-quality affinity matrix. Experimental results on four benchmark datasets show that AWMSCC outperforms seven other advanced clustering algorithms in terms of performance.

**Keywords:** Consistency learning, Tensor nuclear norm (TNN), Adaptive weights, Multi-view subsapce clustering

## 1 Introduction

With the advancement of technology, the demand for information extraction and data mining is growing. In this context, data can be represented in various forms [1-3], such as mathematical symbols, text, and images; it can also represent attributes of objective things, quantities, and relationships between things. These data provide a comprehensive description of entity features from different perspectives. Taking images as an example, researchers refer to sample data that characterize the same entity from different viewpoints as multi-view data. For instance, in order to fully understand the overall outline of a building, it can be viewed from different perspectives such as floor plans, aerial views and sketches. These different types of image information can provide a variety of distinguishable features that help to grasp the overall structural information of things. However, these multi-view data usually exist in high-dimensional space and suffer from problems such as sparse data volume and susceptibility to noise interference. Therefore, how to effectively process multi-view data in high-dimensional spaces has become a hot topic in current investigations.

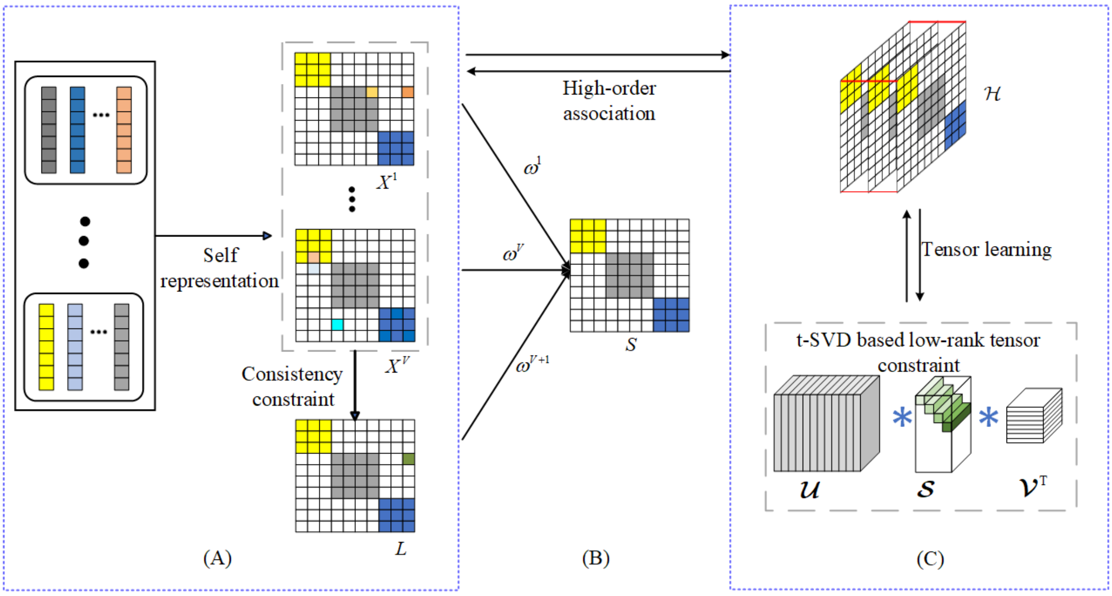
In this context, MSC [4-5] is widely used for its ability to efficiently reduce dimensionality. MSC is based on a reasonable assumption [6] that high-dimensional data are usually distributed in multiple low-dimensional subspaces. By finding the low-dimensional subspace in which the data resides, high-dimensional data can be effectively characterized and data dimensionality reduction can be achieved. Due to its excellent dimensionality reduction capabilities, researchers have proposed many MSC methods. For example, Chen et al. [7] proposed a multi-view low-rank learning method. This method enhances the model's ability to learn complementarities between views by imposing symmetric low-rank constraints on the coefficient matrices. On this basis, Lan et al. [8] proposed a low-rank affinity matrix learning method. The method imposes both symmetry constraints and rank consistency constraints on the matrices, which fully takes into account the global structural information of the data. Furthermore, Tan et al. [9] proposed a sample-level representation method that effectively captures the topological manifold information within views by learning the manifold structure between views and the topological information within each view. Duan et al. [10] introduced a one-step MSC method that directly generates clustering results by integrating self-representation learning, information fusion, and rank equality constraints into a unified framework.

However, the above methods focus more on the structural information within the view and the diversity between views. They often ignore the inherent consistency information between views when multi-view data describes the same sample, which is also crucial for clustering performance. In addition, most of the current methods construct affinity matrices by simply stacking matrices, without considering that the contributions of different views may be different.

To address the above challenges, the authors propose an auto-weighted MSC with consistency learning method, the flowchart is shown in Figure 1. This method not only fully learns the shared features between views but also

*Corresponding Author: Yong Wang; Email: yongwang@ahpu.edu.cn
DOI: https://doi.org/10.70003/160792642025122607003

adaptively assigns appropriate weight information to the views. Firstly, the rank equality constraint is applied to all coefficient matrices, and the shared consistency matrix is further generated by the three-factor decomposition method. On this basis, the coefficient matrices and shared consistency matrix are stacked into a third-order tensor and TNN constraints are used to facilitate the mutual propagation of shared and higher-order information among views, thereby effectively mining consistency and higher-order correlations. Additionally, considering that the contribution of different views may be different, an adaptive weight strategy is employed to assign suitable weight information to all resulting in a higher quality affinity matrix. This article integrates consensus constraints, higher-order information fusion and adaptive weight learning are integrated into a unified framework and optimized by an iterative optimization method to achieve better clustering performance.




(A) Self-representation learning and rank consistency learning. The coefficient matrices are obtained through self-representation of the original matrices, and a shared consistency matrix is derived by applying consistency constraints to the coefficient matrices. (B) View weight learning. Adaptively learns the weight information between different views. (C) Higher-order tensor consistency learning. Concatenates the coefficient matrices and the shared consistency matrix into a third-order tensor and fully learns the higher-order consistency information between views through TNN constraints and t-SVD decomposition.

**Figure 1.** The flowchart of AWMSCC

Main contribution points of this paper are as follows:

- This paper proposes a new MSC method, namely the Auto-weighted MSC with consistency learning (AWMSCC). This method generates a shared consensus matrix through rank equality constraints and learns higher-order consistency information among views by integrating the coefficient matrices with the shared consensus matrix into a third-order tensor.
- The method can adaptively obtain the weight information of each view and effectively solve the problem of contribution imbalance between views.
- An efficient an augmented Lagrange multiplier method (ALM) is designed to solve AWMSCC. Experimental results indicate that AWMSCC outperforms the other seven most advanced algorithms of the species.

## 2 Related Work

Existing multi-view learning methods can be divided into the following four types [11]: (1) Multi-kernel based methods [12], (2) Co-training based methods [13], (3) Graph based methods [14], and (4) Subspace based methods [15]. In the following sections, these four categories of multi-view learning methods will be introduced in detail.

(1) Multi-kernel methods utilize kernel function mapping to make linearly inseparable data separable. Additionally, by learning different kernel functions, diverse feature information can be obtained. For instance, Zhou et al. [16] proposed a multi-kernel graph fusion method. This method obtains multiple base kernels through kernel function mapping, and uses the max-min strategy to adaptively apply weights to the base kernels. Then all basic kernels with weight information are combined into

a composite kernel, and the clustering results are directly obtained through regularization constraints.

(2) Co-training based methods share inter-view consistency information by training a classifier independently for each view and inducing these classifiers to learn from each other. For example, Kumar et al. [17] based on the complementary assumption of spectral clustering, employed an iterative idea where trainers interact to make the clustering results from different views consistent.

(3) Graph-based methods construct a similarity graph matrix by learning the distance between data points, and generate a unified graph matrix by fusing all graph matrix information, which can effectively learn the structural information inside the view. For instance, Nie et al. [18] proposed a completely self-weighting method that uses Laplacian rank constraints to assign appropriate weight information to views to enhance the ability of the model to discriminate the importance of views. Zhan et al. [19] directly generated clustering results by preprocessing all similarity matrices to ensure that high quality initial graphs are obtained, and by imposing rank constraints on the initial graphs.

(4) Subspace-based methods effectively reduce the dimensionality of data by mapping high-dimensional data into low-dimensional subspaces. For instance, Zheng et al. [20] proposed a feature collocation MSC method. The method introduces the concept of cluster-specific corruption and uses dual self-representation to efficiently splice feature information from different views. Zhang et al. [21] proposed a low-rank tensor constraint method, which integrates all the coefficient matrices into a third-order tensor and explains the complementary structural information among different views through tensor low-rank constraints. However, the method cannot adequately explain the structural information of the tensor. On this basis, Xie et al. [22] proposed the tensor singular value decomposition (t-SVD) method, which explicitly characterizes the structural information of the tensor by using the TNN constraint. Since then, tensor learning methods have been widely used in MSC. For example, Du et al. [23] argue that noise in views is diverse. Only by removing both Laplacian and Gaussian noise can we obtain clean affinity matrices and use TNN constraints to fully explore the higher-order structural information of the view. Gao et al. [24] proposed a nuclear norm minimization method, which aims to assign appropriate weight information to singular values according to their sizes.

The four multi-view learning methods mentioned above are specific manifestations in the field of multi-view learning. However, most of these methods fail to fully utilize the inherent structural information between views: some overlook the high-dimensional features of the views, while others do not explore the shared consistency among views. In fact, each view contains information consistent with other views. Therefore, how to effectively utilize the inherent consistency information has become the focus of our research. For our work, by simultaneously applying consistency constraints to all coefficient matrices, we obtain a shared matrix that contains consistency information from all views. This matrix encapsulates the shared information of all views, and combining it with the original coefficient matrices helps us to explore at a higher dimensional level.

**Table 1.** Notations and related descriptions

| Notations | Description |
|---|---|
| A | Matrix |
| $\|\|A\|\|_*$ | The nuclear norm |
| $\|\|A\|\|_{*2,1}$ | The $l_{2,1}$ -norm, $\|A\|_{2,1} = \sum_i \sqrt{\sum_j a_{j,i}^2}$ |
| $\|\|A\|\|_F$ | The Frobenius norm |
| $<A, B>$ | Inner product of A and B |
| $\mathcal{A}$ | 3rd-order tensor |
| $\|\|\mathcal{A}\|\|_{TNN}$ | Tensor norm based on t-SVD |
| $\mathcal{A}*\mathcal{B}$ | t-product of $\mathcal{A}$ and $\mathcal{B}$ |

## 3 The Proposed Approach

This section describes in detail the proposed algorithm, i.e., AWMSCC. In addition, it shows how to integrate consistency learning, high-order feature fusion, and self-weighting strategy into a unified framework. The commonly used symbols and definitions in this article are shown in Table 1.

### 3.1 Objective of AWMSCC

Given data $A\in\mathbb{R}^{n\times m}$, where $m$ denotes the number of samples and $n$ is the number of features. In this paper, we make different data points linearly correlated with each other by self-expression of the multi-view data, which is formulated as follows:

$$\min_{X^v,E^v} \sum_{v=1}^{V} \left\|X^v\right\|_* + \alpha \sum_{v=1}^{V} \left\|E^v\right\|_{2,1} \quad s.t.\ A^v = A^v X^v + E^v, v = 1,2,...,V \tag{1}$$

where $\| \|_*$ denotes the nuclear norm constraint on the coefficient matrices $X^v$ to promote effective approximation of the original rank function information. $\| \|_{2,1}$ is used for processing the noise matrices to achieve sparse handling of noise information in the data, thereby effectively removing noise. In addition, $\alpha$ is the regularization term to increase the generalization ability of the model.

Although self-expression methods can effectively learn the complementary structure between views, they often overlook the inherent consistency information between views. Since different views describe the same samples, the consistency information between views reveals the basic properties shared by the data across different views, meaning that all views should have the same representational structure. Therefore, a rank consistency constraint is applied to all coefficient matrices to maximize the preservation of consensus information among views. The specific formula is as follows:

$$\begin{aligned}&\min_{X^v,E^v}\sum_{v=1}^{V}\left\|X^v\right\|_* + \alpha\sum_{v=1}^{V}\left\|E^v\right\|_{2,1}\\&s.t.\ A^v = A^v X^v + E^v, v=1,2,...,V\\&rank(X^1)=rank(X^2)=\cdots=rank(X^V)\le k\end{aligned} \quad (2)$$

where $k$ denotes the natural upper bound of the coefficient matrices $X^v$, which more accurately approximates the original rank function information. However, this is essentially a soft fixed-rank [25] learning model, and the natural upper bound of the coefficient matrices may be different for different samples, which makes optimization very difficult. Inspired by Lan et al. [8], the problem is effectively solved by further decomposing the coefficient matrix into a three-factor form:

$$\begin{aligned}&\min_{X^v,E^v}\sum_{v=1}^{V}\left\|X^v\right\|_* + \alpha\sum_{v=1}^{V}\left\|E^v\right\|_{2,1}\\&s.t.\ A^v = A^v X^v + E^v, v=1,2,...,V\\&X^v = C^v L R^{v^T}, C^v C^{v^T} = I, R^v R^{v^T} = I\end{aligned} \quad (3)$$

where $C^v$, $R^v \in \mathbb{R}^{m\times k}$ can be considered as the basis vectors in two extended dimensions of coefficient matrices $X^v$. Imposing orthogonal constraints on $C^v$ and $R^v$ respectively can ensure the uniqueness of the solution. $L\in\mathbb{R}^{k\times k}$ denotes the shared consistency matrix. It can make all coefficient matrices share the structure information, which effectively retains the most essential structural features inside the view. In addition, different datasets often require different parameters $k$ to be adjusted for Eq. (3), which can significantly reduce the running efficiency of the algorithm. To avoid this situation, a larger value $k$ is needed, which should at least be greater than the original rank of the matrix. To simplify the parameter selection process, one can directly use the number of samples $m$ instead of $k$. At this point, matrices $C^v$, $R^v$ and $L\in\mathbb{R}^{m\times m}$ are involved.

After obtaining the consistency matrix, in order to ensure that all the coefficient matrices can learn the shared information among views, the coefficient matrices can be spliced with the shared consistency matrix into a tensor. By applying the TNN constraints to the tensor not only promotes the mutual transfer of consistent information between views, but also enables the mining of higher-order correlations between views at a higher level. The specific formula is as follows:

$$\begin{aligned}&\min_{X^v,E^v,\mathcal{H}}\alpha\sum_{v=1}^{V}\left\|E^v\right\|_{2,1} + \beta\left\|\mathcal{H}\right\|_{TNN}\\&s.t.\ A^v = A^v X^v + E^v, v=1,2,...,V, \mathcal{H}=\left\{X^v, L\right\}\\&X^v = C^v L R^{v^T}, C^v C^{v^T} = I, R^v R^{v^T} = I\end{aligned} \quad (4)$$

where $\beta$ is the balance parameter. $\mathcal{H}\in\mathbb{R}^{V+1}$ is formed by stacking the coefficient matrix with the shared consistency matrix.

Additionally, while Eq. (4) adequately exploits the consensus information between views, the final affinity matrix is generated through simple stacking, which does not consider that different views have varying importance to the final clustering results. For example, for a face image, the view generated by the front face often contains more feature information than the image generated by the side face. Therefore, it is necessary to adaptively assign appropriate weight information to different views in order to fully utilize the advantages of the model:

$$\begin{aligned}&\min_{X^v,E^v,\mathcal{H},\omega,S}\alpha\sum_{v=1}^{V}\left\|E^v\right\|_{2,1} + \beta\left\|\mathcal{H}\right\|_{TNN} + \left\|\omega\right\|_2^2\\&\qquad + \gamma\sum_{v=1}^{V+1}\omega^v\left\|S - H^v\right\|_F^2\\&s.t.\ A^v = A^v X^v + E^v, v=1,2,...,V, \mathcal{H}=\left\{X^v, L\right\}\\&\quad X^v = C^v L R^{v^T}, C^v C^{v^T} = I, R^v R^{v^T} = I,\\&\quad \omega^T 1 = 1, \omega^v \ge 0.\end{aligned} \quad (5)$$

where $\omega^v$ is the weight information of the $v$th view, $\left\|\omega\right\|_2^2$ is the smoothing weight constraint, and $\gamma$ is the balancing parameter.

For the AWMSCC algorithm, the first term applies $l_{2,1}$-norm constraint to the noise matrices, effectively removing noise information from the data. The second term is high-order consistency learning, which enables each view to more effectively capture high-order shared information in the data. The third term is a regularization term used to avoid trivial solutions. The fourth term involves view weight learning, which helps better understand the varying impact of different views on clustering performance.

### 3.2 Optimization

Since Eq. (5) is non-convex, it can be solved by constructing the ALM method. The reconstructed objective function is:

$$\begin{aligned}&L\left(X^v, S, E^v, C^v, R^v, L, \mathcal{G}, \omega\right) = \alpha\sum_{v=1}^{V}\left\|E^v\right\|_{2,1}\\&+\beta\left\|\mathcal{G}\right\|_{TNN} + \gamma\sum_{v=1}^{V+1}\omega^v\left\|S-H^v\right\|_F^2 + \left\|\omega\right\|_2^2\\&+\left\langle Q_1^v, A^v - A^v X^v - E^v\right\rangle + \frac{\mu}{2}\left\|A^v - A^v X^v - E^v\right\|_F^2\\&+\left\langle Q_2^v, X^v - C^v L R^{v^T}\right\rangle + \frac{\mu}{2}\left\|X^v - C^v L R^{v^T}\right\|_F^2\\&+\left\langle \mathcal{Q}_3, \mathcal{H}-\mathcal{G}\right\rangle + \frac{\mu}{2}\left\|\mathcal{H}-\mathcal{G}\right\|_F^2\\&s.t.\quad C^v C^{v^T} = I, R^v R^{v^T} = I, \omega^T 1 = 1, \omega^v \ge 0\end{aligned} \quad (6)$$

In this context, $\mu$ is the penalty parameter, $Q_1^v$, $Q_2^v$, $\mathcal{Q}_3$ are the Lagrange multipliers. $\mathcal{G}$ is an auxiliary variable to separate the tensor $\mathcal{H}$, which facilitates easier updating of

Eq. (6). In addition, $X^v$, $S$, $E^v$, $C^v$, $R^v$, $L$, $\mathcal{G}$, $\omega$ are updated separately by an alternating minimization strategy, i.e., fixing other unrelated variables and updating one variable in sequence.

(1) Update $X^v$, Eq. (6) can be expressed as:

$$\begin{aligned}&\min_{X^v}\ \gamma\sum_{v=1}^{V}\omega^v\left\|S-X^v\right\|_F^2\\&+\left\langle Q_1^v, A^v-A^vX^v-E^v\right\rangle+\frac{\mu}{2}\left\|A^v-A^vX^v-E^v\right\|_F^2\\&+\left\langle Q_2^v, X^v-C^vLR^{v^T}\right\rangle+\frac{\mu}{2}\left\|X^v-C^vLR^{v^T}\right\|_F^2\\&+\left\langle Q_3^v, X^v-G^v\right\rangle+\frac{\mu}{2}\left\|X^v-G^v\right\|_F^2\end{aligned}\tag{7}$$

By directly taking the partial derivative of $X^v$ and making Eq. (7) zero, a closed-form solution can be obtained:

$$\begin{aligned}X^v=&\left(2\omega^v\gamma I+\mu A^{v^T}A^v+2\mu I\right)^{-1}\\&(2\omega^v\gamma S+\mu A^{v^T}A^v-\mu A^{v^T}E^v\\&+A^{v^T}Q_1^v-Q_2^v-Q_3^v+\mu C^vLR^{v^T}+\mu G^v)\end{aligned}\tag{8}$$

(2) Update $S$, Eq. (6) can be expressed as:

$$\min_S\sum_{v=1}^{V+1}\omega^v\left\|S-H^v\right\|_F^2\tag{9}$$

Take the partial derivative of parameter $S$ and make Eq. (9) approach zero. The obtained solution is as follows:

$$S=\sum_{v=1}^{V+1}\omega^vH^v\tag{10}$$

(3) Update $E^v$, Eq. (6) can be expressed as:

$$\begin{aligned}&\min_{E^v}\alpha\sum_{v=1}^{V}\left\|E^v\right\|_{2,1}+\left\langle Q_1^v, A^v-A^vX^v-E^v\right\rangle\\&+\frac{\mu}{2}\left\|A^v-A^vX^v-E^v\right\|_F^2\end{aligned}\tag{11}$$

Let $\Phi=A^v-A^vX^v+Q_1^v/\mu$, according to [26], we can get:

$$[E_v]_{:,i}=\begin{cases}\dfrac{\left\|\Phi_{:,i}\right\|_2-\dfrac{\alpha}{\mu}}{\left\|\Phi_{:,i}\right\|_2}\Phi_{:,i}\ , & \left\|\Phi_{:,i}\right\|_2>\dfrac{\alpha}{\mu}\\ 0\ , & otherwise\end{cases}\tag{12}$$

(4) Update $C^v$, Eq. (6) can be expressed as:

$$\min_{C^v}\left\langle Q_2^v, X^v-C^vLR^{v^T}\right\rangle+\frac{\mu}{2}\left\|X^v-C^vLR^{v^T}\right\|_F^2\tag{13}$$

First, we need to find the partial derivative of $C^v$ and get the following form:

$$\min_{C^vC^{v^T}=I}\left\|C^v-\tilde{C}^v\right\|_F^2\tag{14}$$

where $\tilde{C}^v=(Q_2^v(1/\mu)+X^v)R^vL^T$. Eq (14) is an orthogonal Procrustes problem, whose closed-form solution can be obtained according to the [27] as:

$$C^v=\hat{N}^vM^v\tag{15}$$

where $\hat{N}^v$ and $\hat{M}^v$ are the left and the right singular value matrices obtained after performing SVD on $C^v$, respectively.

Similarly, $R^v$ is solved similarly to $C^v$. The subproblem with respect to $R^v$ can be expressed as:

$$\min_{R^vR^{v^T}=I}\left\|R^v-\tilde{R}^v\right\|_F^2\tag{16}$$

where $\tilde{R}^v=(Q_2^v(1/\mu)+X^v)^TC^vL$. In the same way as Eq. (14), its closed-form solution can be obtained as:

$$R^v=\tilde{N}^v\tilde{M}^v\tag{17}$$

where $\tilde{N}^v$ and $\tilde{M}^v$ are the left and the right singular value matrices obtained after performing SVD on $R^v$, respectively.

(5) Update $L$, Eq. (6) can be expressed as:

$$\begin{aligned}&\min_L\gamma\omega\left\|S-L\right\|_F^2+\left\langle Q_2^v, X^v-C^vLR^{v^T}\right\rangle\\&+\frac{\mu}{2}\left\|X^v-C^vLR^{v^T}\right\|_F^2+\left\langle Q_3^{V+1}, L-G^{V+1}\right\rangle\\&+\frac{\mu}{2}\left\|L-G^{V+1}\right\|_F^2\end{aligned}\tag{18}$$

Directly taking the partial derivative of $L$ and setting Eq. (18) to zero, we can obtain the closed-form solution:

$$\begin{aligned}L=&\left(2\gamma\omega I+2\mu I\right)^{-1}\\&\times\frac{1}{V}\sum_{v=1}^{V}\begin{pmatrix}2\gamma\omega S+C^{v^T}Q_2^vR^v+\mu C^{v^T}X^vR^v-Q_3^{V+1}\\+\mu G^{V+1}\end{pmatrix}\end{aligned}\tag{19}$$

(6) Update $\mathcal{G}$, Eq. (6) can be expressed as:

$$\min_{\mathcal{G}} \beta \|\mathcal{G}\|_{TNN} + \langle \mathcal{Q}, \mathcal{H} - \mathcal{G} \rangle + \frac{\mu}{2} \|\mathcal{H} - \mathcal{G}\|_F^2 \tag{20}$$

The solution of Eq. (20) can be obtained from [28].

(7) Update $\omega$, Eq. (6) can be expressed as:

$$\begin{aligned} &\min_{\omega} \gamma \sum_{v=1}^{V+1} \omega^v \|S - H^v\|_F^2 + \|\omega\|_2^2 \\ &s.t.\ \omega^T 1 = 1, \omega^v \geq 0 \end{aligned} \tag{21}$$

Eq. (21) can be solved using the quadprog function in the MATLAB toolbox.

(8) Update $Q_1^v$, $Q_2^v$, $\mathcal{Q}$, $\mu$:

$$\begin{cases} Q_1^v = Q_1^v + \mu\left(A^v - A^v X^v - E^v\right) \\ Q_2^v = Q_2^v + \mu\left(X^v - C^v L R^{v^T}\right) \\ \mathcal{Q} = \mathcal{Q} + \mu(\mathcal{H} - \mathcal{G}) \\ \mu = \min(\eta\mu, \mu_{\max}) \end{cases} \tag{22}$$

Algorithm 1 summarizes the complete optimization process of AWMSCC.

**Algorithm 1.** AWMSCC

**Input:** Multi-view data $A^v$, parameter $\alpha$, $\beta$ and $\gamma$.
**Output:** Clustering result.
**Initialized:** $Q_1^v = Q_2^v = \mathcal{Q}_3 = X^v = C^v = E^v = R^v = 0$, $L = \mathcal{G} = 0$, $\omega^v = 1/(V+1)$, $\mu = 10^{-5}$, $\mu_{\max} = 10^{10}$, $\eta = 2$, $\varepsilon = 0.001$.

```
While not converged do
    Update X^v by Eq. (8);
    Update S by Eq. (10);
    Update E^v by Eq. (12);
    Update C^v and R^v by Eq. (15) and (17);
    Update L by Eq. (19);
    Update G by Eq. (20);
    Update ω by Eq.(21);
    Update Q_1^v, Q_2^v, Q_3 by Eq. (22);
end while
Update μ by Eq. (22);
Perform spectral clustering on S.
```

# 4 Experiment

### 4.1 Datasets

This article selects four public datasets, Table 2 provides a visual summary of the four datasets. Here is a brief overview of each dataset:

**BBCSport:** This dataset comes from the BBC and contains 544 sports news documents. Two different angles are extracted as view information.

**BBC4VIEW:** This dataset also comes from the BBC and includes 685 subject documents. Four different features are extracted for each document.

**NGs:** This dataset contains 500 newsgroup samples and has information from three different views of the samples collected through three distinct collection methods.

**ORL:** This dataset collects facial images of 40 different individuals. Each person has 10 faces, totaling 400 images. Three distinct features are extracted as view information.

### 4.2 Comparison Methods

To demonstrate the superiority of the AWMSCC algorithm, it will be compared with seven other advanced multi-view clustering algorithms. Additionally, parameter tuning will be performed for each clustering algorithm to achieve optimal performance. The following is a brief description of the compared algorithms:

**SMVSC** [29]: This algorithm constructs a unified framework by integrating anchor points and graph learning methods, more accurately describing the true distribution of view data.

**LT-MSC** [21]: LT-MSC learns higher-order correlations between views through low-rank tensor constraints.

**ETLMSC** [30]: This method introduces probabilistic transfer tensor to subspace learning for the first time, which effectively reduces the running time of the algorithm.

**t-SVD-MSC** [22]: This method applies t-SVD method, providing a more intuitive way to learn higher-order information between views.

**LSGMC** [8]: This method explores view consistency through symmetric constraints and rank consistency constraint method.

**HLR-MVS** [31]: This method uses hypergraphs to capture view correlations and local flow structures.

**DV-MSC** [11]: This method divides the view features into two dimensions and learns the complementarity and high-order consistency information of views from low to high at two different levels.

**Table 2.** Dataset description

| Datasets | Objective | Size | Clusters | Dimensions |
|---|---|---|---|---|
| BBCSport | Text | 544 | 5 | [3183, 3203] |
| BBC4View | Text | 685 | 5 | [4659, 4633, 4665, 4684] |
| NGs | Text | 500 | 5 | [2000, 2000, 2000] |
| ORL | Face | 400 | 40 | [4096, 3304, 6750] |

### 4.3 Evaluation Metrics

To comprehensively assess the performance of AWMSCC, six evaluation metrics are selected to measure the model. These six metrics are ACC, NMI, F-Score, Precision, AR and Recall.

### 4.4 Performance Comparison

During the experimental stage, optimal parameters were selected for each algorithm. The clustering performance comparison results of AWMSCC with the other seven algorithms are detailed in Table 3. In these tables, the bold values indicate optimal performance, the underlined values suboptimal performance.

**Table 3.** Experimental results on four datasets

| Dataset | Methods | NMI | ACC | AR | F-score | Precision | Recall |
|---|---|---|---|---|---|---|---|
| BBCSport | SMVSC | 0.7072 | 0.8768 | 0.7118 | 0.7789 | 0.7978 | 0.7609 |
| | LT-MSC | 0.3223 | 0.5638 | 0.2511 | 0.4905 | 0.3657 | 0.7775 |
| | ETLMSC | 0.9688 | 0.9890 | 0.9803 | 0.9850 | 0.9895 | 0.9806 |
| | t-SVD-MSC | 0.9049 | 0.9706 | 0.9172 | 0.9371 | 0.9313 | 0.9429 |
| | LSGMC | 0.9171 | 0.9761 | 0.9378 | 0.9525 | 0.9602 | 0.9449 |
| | HLR-MVS | 0.9158 | 0.9743 | 0.9334 | 0.9492 | 0.9504 | 0.9481 |
| | DV-MSC | 0.9884 | 0.9963 | 0.9932 | 0.9948 | 0.9965 | 0.9921 |
| | **AWMSCC** | **1.0000** | **1.0000** | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| BBC4View | SMVSC | 0.6939 | 0.8759 | 0.7198 | 0.7855 | 0.7854 | 0.7855 |
| | LT-MSC | 0.7566 | 0.9109 | 0.7913 | 0.8404 | 0.8381 | 0.8427 |
| | ETLMSC | 0.5412 | 0.6623 | 0.4031 | 0.5338 | 0.5717 | 0.5006 |
| | t-SVD-MSC | 0.9233 | 0.9635 | 0.9150 | 0.9344 | 0.9609 | 0.9093 |
| | LSGMC | 0.9275 | 0.8380 | 0.7807 | 0.7860 | 0.7391 | 0.8396 |
| | HLR-MVS | 0.9041 | 0.9504 | 0.9002 | 0.9229 | 0.9508 | 0.8967 |
| | DV-MSC | 0.8858 | 0.9650 | 0.9140 | 0.9340 | 0.9406 | 0.9275 |
| | AWMSCC | **0.9539** | **0.9810** | **0.9628** | **0.9714** | **0.9826** | **0.9605** |
| NGs | SMVSC | 0.7358 | 0.8680 | 0.7208 | 0.7771 | 0.7645 | 0.8442 |
| | LT-MSC | 0.9652 | 0.9900 | 0.9750 | 0.9799 | 0.9798 | 0.9801 |
| | ETLMSC | 0.3920 | 0.5190 | 0.2671 | 0.4374 | 0.3734 | 0.5280 |
| | t-SVD-MSC | 0.9722 | 0.9920 | 0.9800 | 0.9840 | 0.9838 | 0.9841 |
| | LSGMC | 0.8507 | 0.9460 | 0.8705 | 0.8962 | 0.8954 | 0.8971 |
| | HLR-MVS | 0.9780 | 0.9371 | 0.9465 | 0.9571 | 0.9569 | 0.9573 |
| | DV-MSC | 0.9686 | 0.9900 | 0.9750 | 0.9799 | 0.9797 | 0.9802 |
| | **AWMSCC** | **1.0000** | **1.0000** | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| ORL | SMVSC | 0.9204 | 0.8075 | 0.7603 | 0.7661 | 0.7197 | 0.8188 |
| | LT-MSC | 0.9151 | 0.8130 | 0.7518 | 0.7578 | 0.7147 | 0.8067 |
| | ETLMSC | 0.9868 | 0.9503 | 0.9467 | 0.9479 | 0.9242 | 0.9731 |
| | t-SVD-MSC | 0.9671 | 0.9150 | 0.8979 | 0.9002 | 0.8792 | 0.9222 |
| | LSGMC | 0.9322 | 0.7980 | 0.7578 | 0.7639 | 0.6873 | 0.8611 |
| | HLR-MVS | 0.9839 | 0.9300 | 0.9280 | 0.9297 | 0.8926 | 0.9700 |
| | DV-MSC | 0.9797 | 0.9175 | 0.9123 | 0.9144 | 0.8743 | 0.9583 |
| | **AWMSCC** | **0.9929** | **0.9625** | **0.9652** | **0.9660** | **0.9467** | **0.9861** |

AWMSCC achieved the best performance across all four benchmark datasets, showing a significant improvement compared to other algorithms. Specifically, on the BBCSport and NGs datasets, all clustering performance metrics of AWMSCC were 1, indicating exceptional performance without any misclassified points. On the ORL dataset, the clustering performance of AWMSCC is close to that of ETLMSC and HLR-MVS. This may be because the dataset contains less feature information and is relatively easy to process. On the BBC4View dataset, AWMSCC achieved excellent performance, and compared with the second best t-SVD-MSC, the clustering performance was improved by 3.31%, 1.81%, 5.22%, 4.07%, 2.25%, and 5.6%, respectively.

The matrix-based methods, LSGMC and SMVSC, performed poorly on the four datasets. The primary reason is that they only utilized the matrix nuclear norm to capture the structural information of views, without fully considering the higher-order correlations between views. HLR-MVS learns view structure information by hyper-Laplace regularization, but its clustering performance is average. This is because while it learns local structural information within views, it ignores the inherent consistency information between views. DV-

MSC is a recently proposed dual-view structure learning method that performs better on most datasets, but still lagged behind AWMSCC. This is mainly because DV-MSC takes into account both the consistency and high-order correlation between views, but ignores the different effects of different views on the final result. ETLMSC, t-SVD-MSC and the proposed AWMSCC are all tensor-based methods, but AWMSCC outperforms these two methods on all datasets. This is because AWMSCC not only learns the high-order structural information between views through TNN constraints, but also fully considers the inherent consistency information between views and the difference in the contribution of each view to clustering performance. AWMSCC effectively solves the problems of the above methods by simultaneously learning the high-order consistency between views and the problem of view contribution mismatch, and achieves the best performance in the experiment.

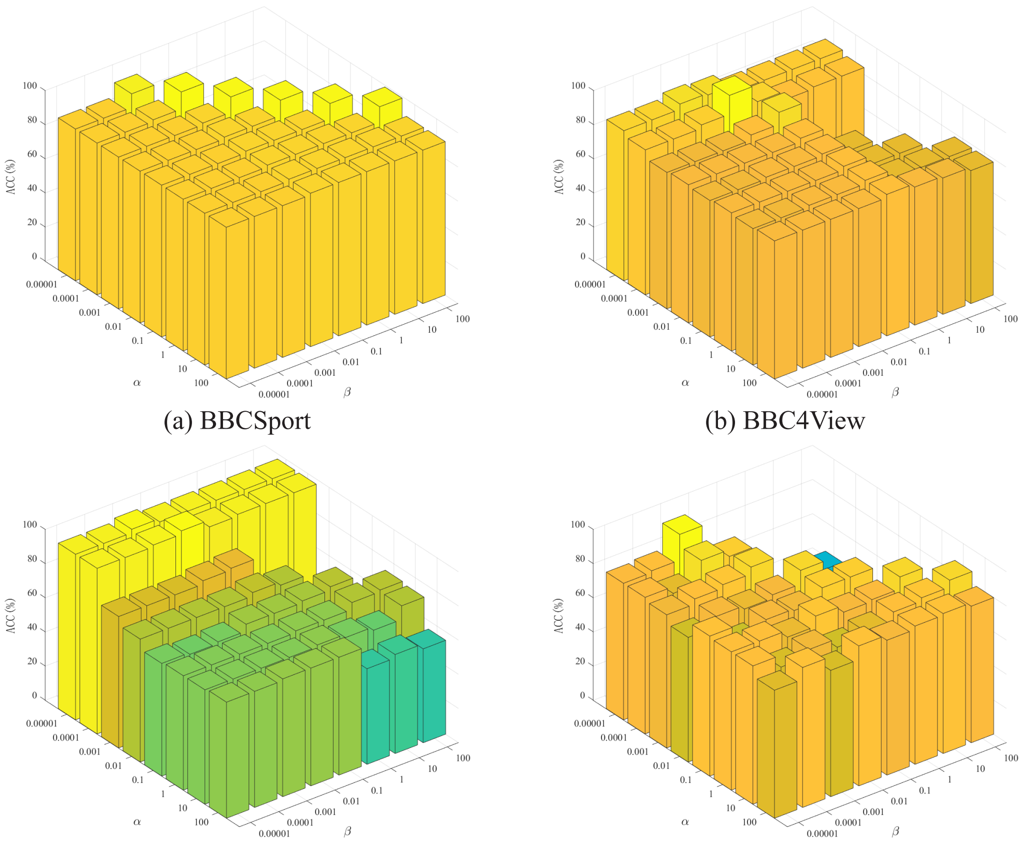
### 4.5 Performance Comparison

The AWMSCC algorithm requires manual adjustment of three parameters: $\alpha$, $\beta$, and $\gamma$. These parameters are all selected within the range [0.00001, 0.0001, ..., 100]. Figure 2 illustrates the sensitivity of parameters $\alpha$ and $\beta$ in AWMSCC, where $\gamma$ is fixed as the optimal value. As can be seen from Figure 2, selecting different values for $\alpha$ and $\beta$ can maintain stable clustering performance, indicating that the algorithm has a certain robustness. In Figure 3, the effect of $\gamma$ on the clustering performance is explored by keeping the values of parameters $\alpha$ and $\beta$ constant. From Figure 3, it can be seen that different values of $\gamma$ have a greater impact on the clustering performance, which implies that the contribution of views is crucial for clustering performance. Additionally, the optimal parameters corresponding to $\alpha$, $\beta$ and $\gamma$ for datasets BBCSport, BBC4View, NGs and ORL are as follows: (0.0001,0.01,0.00001), (0.0001,0.01,0.1), (0.0001,0.01,0.1) and (0.00001,0.001,0.01).

### 4.6 Convergence Analysis

The convergence of AWMSCC is discussed in this section. Figure 4 shows the convergence plots of AWMSCC on four different datasets. Reconstruction error is defined as follows:

$$\max\begin{cases} error1=\dfrac{1}{V}\sum\limits_{v=1}^{V}\left\|A^v-A^vX^v-E^v\right\|_\infty \\ error2=\dfrac{1}{V}\sum\limits_{v=1}^{V}\left\|X^v-C^vLR^{v^T}\right\|_\infty \\ error3=\left\|\mathcal{H}-\mathcal{G}\right\|_\infty \end{cases} \quad (23)$$




(a) BBCSport (b) BBC4View

(c) NGs (d) ORL

**Figure 2.** Parameter analysis on all dataset

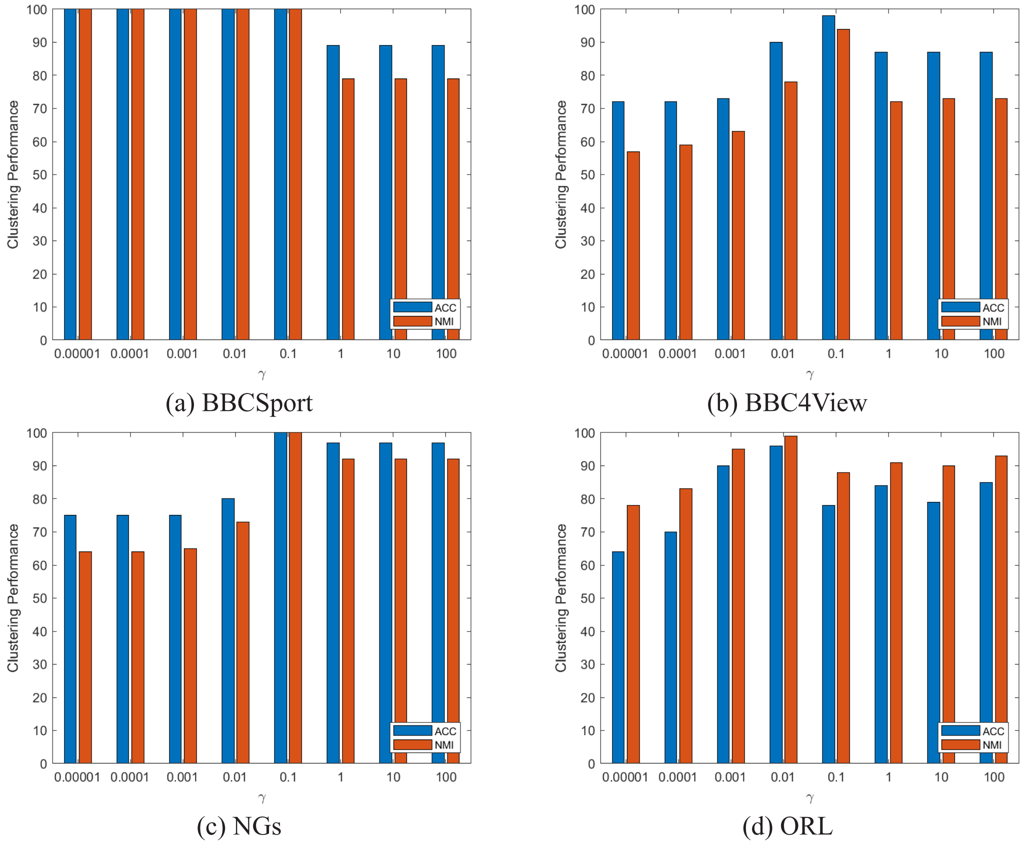



**Figure 3.** Parameter analysis of $\gamma$

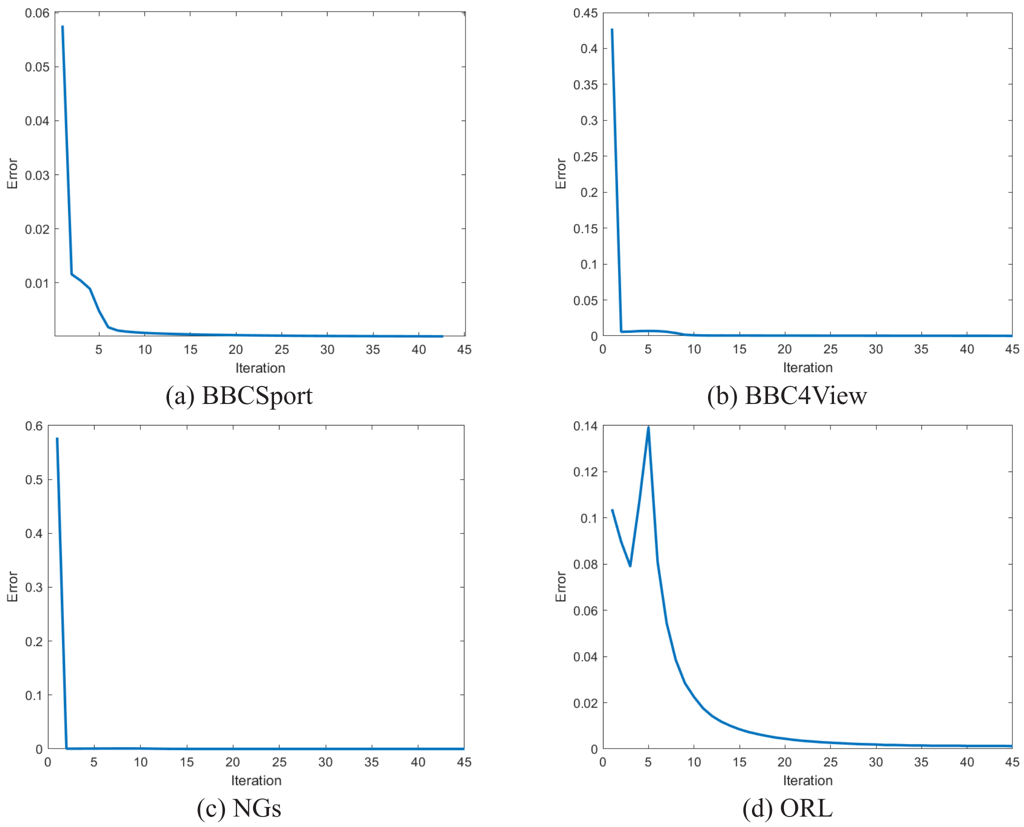



**Figure 4.** Convergence results of AWMSCC

As can be seen in Figure 4, after about 40-45 iterations, the error converges to 0. Based on these observations, it is shown that AWMSCC has excellent convergence.

# 5 Conclusion

In this paper, we propose a new MSC method, namely AWMSCC, which not only learns the consensus information among views at a high-dimensional level, but also fully takes into account the difference in importance of different views for clustering. First, coefficient matrices are obtained by self-representing all original matrices. Next, rank consistency constraints are imposed on these coefficient matrices and a three-factor decomposition is performed to generate shared consistency matrices. To ensure that all coefficient matrices learn the consistency information between views, the coefficient matrices are further spliced into a tensor with the shared consistency matrix and TNN constraints are imposed so that the consistency information can be propagated to each other at a high dimensional level. Additionally, a self-weighting strategy among views assigns appropriate weights to each view. Although AWMSCC performs well in terms of clustering performance, the higher computational complexity and longer runtime are issues that cannot be ignored. In the future, we plan to adopt the Markov chain method to replace self-representation learning in order to reduce the overall computational complexity of the algorithm. In addition, we also intend to incorporate the concept of one-step learning into AWMSCC with the aim of directly generating the final clustering results, thus further improving the efficiency.

# Acknowledgements

This work was supported by National Natural Science Foundation of China under project (Grant No. 62002084, 61976005), the University Synergy Innovation Program of Anhui Province (Grant No. GXXT-2022-047), the Key Project Anhui New Era Education Provincial Quality Engineering (Graduate Education) (Grant No. 2023jyjxggyjY153), Safety-Critical Software Key Laboratory Research Program (No. NJ2023031).

# References

[1] U. Fang, M. Li, J. Li, L. Gao, T. Jia, Y. Zhang, A comprehensive survey on multi-view clustering, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 12, pp. 12350-12368, December, 2023.

[2] B. B. JV, J. M. Philip, C. KT, A. KP, A Framework for Analyzing the Context of Discussion in Crowd Clusters, *International Journal of Performability Engineering*, Vol. 20, No. 4, pp. 224-231, April, 2024.

[3] M. Singh, G. Jindal, A. Oberoi, R. Dhangar, Improving Crime Detection Through Geo-MDA: A Hybrid Linear Regression Approach in Data Mining, *International Journal of Performability Engineering*, Vol. 20, No.8, pp. 469-477, August, 2024.

[4] G. A. Khan, J. Hu, T. Li, B. Diallo, S. Du, Multi-view subspace clustering for learning joint representation via low-rank sparse representation, *Applied Intelligence*, Vol. 53, No. 19, pp. 22511-22530, October, 2023.

[5] L. Li, Z. Wan, H. He, Incomplete multi-view clustering with joint partition and graph learning, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 1, pp. 589-602, January, 2023.

[6] X. Gao, X. Ma, W. Zhang, J. Huang, H. Li, Y. Li, J. Cui, Multi-view clustering with self-representation and structural constraint, *IEEE transactions on big data*, Vol. 8, No. 4, pp. 882-893, August, 2022.

[7] J. Chen, S. Yang, H. Mao, C. Fahy, Multiview subspace clustering using low-rank representation, *IEEE Transactions on Cybernetics*, Vol. 52, No. 11, pp. 12364-12378, November, 2022.

[8] W. Lan, T. Yang, Q. Chen, S. Zhang, Y. Dong, H. Zhou, Y. Pan, Multiview subspace clustering via low-rank symmetric affinity graph, *IEEE Transactions on Neural Networks and Learning Systems*,Vol. 35, No. 8, pp. 11382-11395, August, 2024.

[9] Y. Tan, Y. Liu, S. Huang, W. Feng, J. Lv, Sample-level multi-view graph clustering, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, BC, Canada, 2023, pp. 23966-23975.

[10] Y. Duan, H. Yuan, L. L. Lai, B. He, Multi-view subspace clustering with local and global information, *2021 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, Adelaide, Australia, 2021, pp. 1-6.

[11] S. Lan, Q. Zheng, Y. Yu, Double-level view-correlation multi-view subspace clustering, *Knowledge-Based Systems*, Vol. 284, No. 1, Article No. 111271, January, 2024.

[12] A. Monney, Y. Zhan, Z. Jiang, B. Benuwa, A multi-kernel method of measuring adaptive similarity for spectral clustering, *Expert Systems with Applications*, Vol. 159, No. 1, Article No. 113570, November, 2020.

[13] J. Liu, X. Liu, Y. Yang, X. Guo, M. Kloft, L. He, Multiview subspace clustering via co-training robust data representation, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 33, No. 10, pp. 5177-5189, October, 2022.

[14] W. Feng, Z. Wang, T. Xiao, M. Yang, Adaptive weighted dictionary representation using anchor graph for subspace clustering, *Pattern Recognition*, Vol. 151, No. 1, Article No. 110350, July, 2024.

[15] B. Pan, C. Li, H. Che, Error-robust multi-view subspace clustering with nonconvex low-rank tensor approximation and hyper-Laplacian graph embedding, *Engineering Applications of Artificial Intelligence*, Vol. 133, No. 1, Article No. 108274, July, 2024.

[16] B. Zhou, W. Liu, W. Zhang, Z. Lu, Q. Tan, Multi-kernel graph fusion for spectral clustering, *Information Processing & Management*, Vol. 59, No. 5, Article No. 103003, September, 2022.

[17] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, *Proceedings of the 28th international conference on machine learning (ICML-11)*, Madison, WI, USA, 2011, pp. 393-400.

[18] F. Nie, X. Wang, M. Jordan, H. Huang, The constrained laplacian rank algorithm for graph-based clustering, *Proceedings of the AAAI conference on artificial intelligence*, Phoenix, Arizona, USA, 2016, pp. 1969-1976.

[19] K. Zhan, C. Zhang, J. Guan, J. Wang, Graph learning for multiview clustering, *IEEE transactions on cybernetics*,

Vol. 48, No. 10, pp. 2887-2895, October, 2018.
[20] Q. Zheng, J. Zhu, Z. Li, S. Pang, J. Wang, Y. Li, Feature concatenation multi-view subspace clustering, *Neurocomputing*, Vol. 379, pp. 89-102, February, 2020.
[21] C. Zhang, H. Fu, S. Liu, G. Liu, X. Cao, Low-rank tensor constrained multiview subspace clustering, *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, 2015, pp. 1582-1590.
[22] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, Y. Qu, On unifying multi-view self-representations for clustering by tensor multi-rank minimization, *International Journal of Computer Vision*, Vol. 126, No. 11, pp. 1157-1179, November, 2018.
[23] S. Du, B. Liu, G. Shan, Y. Shi, W. Wang, Enhanced tensor low-rank representation for clustering and denoising, *Knowledge-Based Systems*, Vol. 243, No. 1, Article No. 108468, May, 2022.
[24] Q. Gao, W. Xia, Z. Wan, D. Xie, P. Zhang, Tensor-SVD based graph learning for multi-view subspace clustering, *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, New York, USA, 2020, pp. 3930-3937.
[25] H. Chen, Y. Sun, J. Gao, Y. Hu, B. Yin, Fast optimization algorithm on Riemannian manifolds and its application in low-rank learning, *Neurocomputing*, Vol. 291, pp. 59-70, May, 2018.
[26] Z. Ding, Y. Fu, Low-rank common subspace for multi-view learning, *2014 IEEE international conference on Data Mining*, Shenzhen, China, 2014, pp. 110-119.
[27] P. H. Schönemann, A generalized solution of the orthogonal procrustes problem, *Psychometrika*, Vol. 31, No. 1, pp. 1-10, March, 1966.
[28] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 35, No. 1, pp. 171-184, January, 2013.
[29] M. Sun, P. Zhang, S. Wang, S. Zhou, W. Tu, X. Liu, E. Zhu, C. Wang, Scalable multi-view subspace clustering with unified anchors, *Proceedings of the 29th ACM international conference on multimedia*, Virtual Event, China, 2021, pp. 3528-3536.
[30] J. Wu, Z. Lin, H. Zha, Essential tensor learning for multi-view spectral clustering, *IEEE Transactions on Image Processing*, Vol. 28, No. 12, pp. 5910-5922, December, 2019.
[31] Y. Xie, W. Zhang, Y. Qu, L. Dai, D. Tao, Hyper-Laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning, *IEEE transactions on cybernetics*, Vol. 50, No. 2, pp. 572-586, February, 2020.

## Biographies

**Ao Li** received the B.S. degrees in computer science from Anhui Polytechnic University. He is currently pursuing the M.S. degree in computer technology at Anhui Polytechnic University, China. His current research interests include Machine learning, pattern recognition, and software engineering.

**Yong Wang** received his B.S. and M.S. degrees in computer science from Anhui Polytechnic University, and he received his Ph.D. degree in computer science and technology from Nanjing University of Aeronautics and Astronautics. His current research interests include software testing, fault localization, and program debugging.

**Lihao Yang** received the B.S. degrees in computer science from Anhui Institute of Information Technology. He is currently studying for his M.S. degree in computer technology at Anhui Polytechnic University, China. His current research interests include software testing, fault localization, and program debugging.

**Shuai Meng** received the B.S. degrees in computer science from Anhui Polytechnic University and continued the M.S. degree in computer technology at Anhui Polytechnic University, China. His current research interests include software testing, fault localization, and program debugging.