Federated Multi-View Clustering via Tensor Factorization

Wei Feng[1] , **Zhenwei Wu**[1] , **Qianqian Wang** [2*] , **Bo Dong** [3] , **Zhiqiang Tao**[4] and **Quanxue Gao**[2]
[1]School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
[2]School of Telecommunications Engineering, Xidian University, Xi'an, China
[3]School of Continuing Education, Xi'an Jiaotong University, Xi'an, China
[4]School of Information, Rochester Institute of Technology, NY, USA;
weifeng.ft@xjtu.edu.cn, wzw3213151007@stu.xjtu.edu.cn, qqwang@xidian.edu.cn,
dong.bo@mail.xjtu.edu.cn, zqtaomail@gmail.com, qxgao@xidian.edu.cn

Abstract

Multi-view clustering is an effective method to process massive unlabeled multi-view data. Since data of different views may be collected and held by different parties, it becomes impractical to train a multi-view clustering model in a centralized way, for the sake of privacy. However, federated multi-view clustering is challenging because multi-view learning has to consider the complementary and consistent information between each view distributed across different clients. For another, efficiency is highly expected in federated scenarios. Therefore, we propose a novel federated multi-view clustering method with tensor factorization(TensorFMVC), which is built based on K-means and hence is more efficient. Besides, TensorFMVC avoids initializing centroids to address the performance degradation of K-means due to its sensitivity to centroid initialization. A three-order tensor stacked by cluster assignment matrices is introduced to exploit the complementary information and spatial structure of different views. Furthermore, we divide the optimization into several subproblems and develop a federated optimization approach to support cooperative model training. Extensive experiments on several datasets demonstrate that our proposed method exhibits superior performance in federated multi-view clustering.

1 Introduction

Multi-view clustering has been an effective unsupervised method to deal with multi-view data in recent years because it fully explores the complementary and consistent information between views. Existing multi-view clustering methods can be roughly classified into two categories [Fang *et al.*, 2023b]: deep multi-view clustering [Liu *et al.*, 2023; Yan *et al.*, 2023; Xu *et al.*, 2022] and heuristic multi-view clustering [Xing *et al.*, 2023; Cai *et al.*, 2023]. For example, [Xu *et al.*, 2023] unties the entanglement between useful and meaningless information to learn disentangled and interpretable representations. [Chen *et al.*, 2023] introduces federated learning to deep multi-view clustering to deal with security concerns by exploiting complementary cluster structures across clients and sharing non-sensitive information to avoid data leakage. Despite the outstanding performance of deep multi-view clustering, the high computation limits its real-world application. Therefore, heuristic multi-view clustering methods still show their application significance in practice. Among heuristic multi-view clustering methods, multi-view K-means methods have drawn considerable attention due to their low complexity and high efficiency [Yang and Sinaga, 2019]. However, most methods adopt centralized settings, while real-world multi-view data may well be collected and held by different clients who expect their data to be secret.

To address this issue, federated multi-view K-means were developed inspired by federated learning [Dong *et al.*, 2022]. For example, [Huang *et al.*, 2022] built a federated multi-view clustering based on non-negative matrix factorization (NMF) and K-means; [Hu *et al.*, 2023] proposed a federated multi-view fuzzy K-means with efficiency improvement. Nonetheless, they leveraged randomly initialized cluster centroids and thus suffered from unstable clustering performance. The intrinsic weaknesses of the underlying K-means greatly limit the practical performance of these federated multi-view clustering methods based on K-means. Although this issue has been widely discussed by existing works [Pei *et al.*, 2022], it is nontrivial to apply them directly in federated scenarios due to the distributed properties and privacy preservation requirements.

To overcome these weaknesses, we propose a novel Federated Multi-View Clustering method via Tensor Factorization (TensorFMVC). The proposed method mitigates the sensitivity of K-means to centroid initialization by adopting centerless K-means. Besides, it employs an anchor-based affinity matrix rather than Euclidean distance, such that it effectively reduces the computational complexity and alleviates the weaknesses of traditional K-means that it cannot handle linearly non-separable data. Besides, inspired by tensor-based methods, we introduce the Schatten p-norm to extract the complementary information shared by all views. To validate the proposed method in federated scenarios, we develop a distributed optimization algorithm, enabling the model to be optimized collaboratively. Specifically, the main contributions of our work are summarized as follows:

- We propose a federated centerless multi-view K-means that can solve the initialization problem of traditional federated multi-view K-means. By replacing the Eu-

clidean distance with an anchor-based affinity graph, the proposed method can better handle linearly non-separable data and well reduce the computational complexity;

- We build a three-order tensor by stacking the indicator matrix of each view and introduce a tensor Schatten p-norm based regularizer to explore the complementary information, which can be solved by tensor singular value decomposition (t-SVD);
- We develop an optimization algorithm to support that multiple clients could collaboratively train the proposed model without exchanging any private multi-view data;
- Extensive experiments and analyses with several methods are conducted on several multi-view datasets, and the experimental results indicate the effectiveness and superiority of our Tensor-FMVC.

2 Related Work

2.1 Federated Multi-View K-means

Multi-view K-means clustering is widely used due to its low computational cost and easily parallelized process [Cai *et al.*, 2013]. For example, [Huang *et al.*, 2018] utilized the capped-norm loss as the objective in multi-view K-means clustering to remove the negative effect of the outliers. For a similar purpose, [Chen *et al.*, 2020] considered both attribution outliers and class outliers and simultaneously performed clustering and outliers detection on the multi-view data to remove the negative influence of outliers. MVASM [Han *et al.*, 2020] utilized consistency, independence, and complementary information between views by a common membership matrix with the proper sparseness. Federated multi-view clustering enables multiple clients to construct a global model with distributed multi-view data, which satisfies the requirements of data privacy preservation in the federal scenario [Feng and Yu, 2020]. [Huang *et al.*, 2022] proposed an efficient multi-view clustering model with NMF and K-means (FedMVL), which can handle the stragglers and fault tolerance. Similarly, [Hu *et al.*, 2023] realized a federated multi-view fuzzy C-means (FedMVFCM).

2.2 Schatten p-Norm

Schatten p-norm is defined on the singular value of a matrix and helps to exploit the complementary information in multi-view learning [Kong *et al.*, 2018; Wang *et al.*, 2023], which has been widely applied in multi-view clustering [Zhang *et al.*, 2019]. [Li *et al.*, 2023] introduced auto-weighted tensor Schatten p-norm to multi-view clustering and it can also automatically and appropriately shrink singular values to fully capture spatial structure in the graph tensor. [Zhao *et al.*, 2022] employed the Schatten p-norm to minimize the divergence between graphs of different views to learn optimal clustering results, and an adaptive weighted strategy was adopted to adjust the contribution of each view. Similar work was also proposed in [Lu *et al.*, 2023]. For easier presentation, we present the preliminaries of Schatten p-norm in this section [Liu *et al.*, 2019].

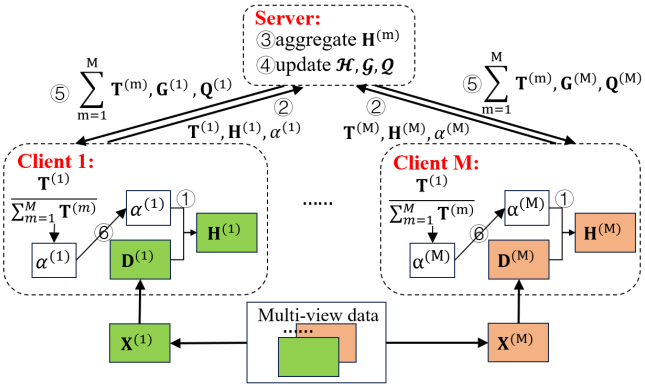

Figure 1: The general framework of the proposed TensorFMVC, and for simplification, $\mathbf{T}^{(\mathbf{m})} = Tr(\mathbf{H}^{(m)^T}\mathbf{D}^{(m)}\mathbf{H}^{(m)})^{\frac{1}{1-r}}$.

Definition 1 (Tensor Singular Value Decomposition, t-SVD)**.** *Give a tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then we have its t-SVD:*

$$\mathcal{M} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T \tag{1}$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are both orthogonal tensors, and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is diagonal tensor. Moreover, all frontal slices of $\mathcal{S}$ are diagonal matrixes, and the i-th singular value of k-th frontal slices of $\mathcal{S}$ is denoted as $\mathcal{S}^k(i)$.

Definition 2. *The tensor Schatten p-norm for tensor $\mathcal{M}$ is defined as below:*

$$||\mathcal{M}||_{S_p} = \left(\sum_{i=1}^{\min(n_1,n_2)} \sum_{k=1}^{n_3} \left(\mathcal{S}^k(i)\right)^p \right)^{\frac{1}{p}} \tag{2}$$

$\mathcal{S}^k(i)$ is the i-th singular value of matrix $\mathcal{S}^k$. with $0 \leq p \leq 1$, the tensor Schatten p-norm is better rank approximation than other norms.

Definition 3. *For $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the function*

$$\min_{\mathcal{B}} \tau ||\mathcal{B}||_{S_p}^p + \frac{1}{2}||\mathcal{B} - \mathcal{C}||_F^2 \tag{3}$$

has the optimal solution given by

$$\mathcal{B}^* = ifft\left(\mathcal{U} * \mathcal{D}_{\tau,p}(\mathcal{C}) * \mathcal{V}^T\right) \tag{4}$$

where $\mathcal{U}$ and $\mathcal{V}$ are gotten from t-SVD of $\mathcal{C}$, i.e. $\mathcal{C} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$, and $\mathcal{D}_{\tau,p}\left(\mathbf{C}^{(m)}\right) = diag\left(\zeta\left(\mathbf{C}^{(\mathbf{m})}\right)\right)$, where $\zeta\left(\mathbf{C}^{(\mathbf{m})}\right) = GST\left(\sigma\left(\mathbf{S}^{(\mathbf{m})}, \tau, p\right)\right)$. The GST algorithm is introduced in [Gao et al.*, 2020].*

3 Method

3.1 Problem Statement

In a federated scenarios, there is a server $\mathcal{S}$ along with M clients, and the multi-view data is denoted as $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(M)}\}$, where M is the number of views. The data distributed on client m is denoted as $\mathbf{X}^{(m)} \in \mathbb{R}^{N \times d^{(m)}} (m = 1, 2, \cdots, M)$ where N is the sample number and $d^{(m)}$ is the feature dimension. The proposed method tries to leverage the local data of all clients to learn the cluster assignment in the federated scenarios.

3.2 Proposed Method

Taking the m-th view as an example, the single-view K-means can be expressed as:

$$\min_{A_1,,\cdots,A_K} = \sum_{k=1}^{K} \sum_{\mathbf{x}_j^{(m)} \in A_k} ||\mathbf{x}_j^{(m)} - \mathbf{m}_k^{(m)}||_2^2 \tag{5}$$

where $A_1, A_2, \cdot \cdot \cdot, A_C$ represent the C clusters and $\mathbf{m}_k^{(m)} = \sum_{\mathbf{x}_j \in A_C} \frac{\mathbf{x}_j}{n_k}$ is the centroid of A_k.

K-means requires an initialization of centroids, which greatly influences the clustering performance. We follow the work [Lu *et al.*, 2023] to build a centerless K-means. Suppose $\mathbf{D^{(m)}}$ is the distance matrix of the m-th view $\mathbf{H}^{(m)} = [\mathbf{h}_1^{(m)}, \mathbf{h}_2^{(m)}, \cdot \cdot \cdot, \mathbf{h}_N^{(m)}]^T \in \mathbb{R}^{N \times K}$ is the indicator matrix, where $\mathbf{h}_{ij}^{(m)} = 1$ indicates $\mathbf{x}_i^{(m)} \in A_j$ and $\mathbf{h}_i^{(m)}$ is a one-hot vector. Then, **Eq**. (5) can be simplified into:

$$\min_{\mathbf{H}^{(m)}} Tr\left(\left({\mathbf{H}^{(m)}}^T \mathbf{H}^{(m)})^{-1} {\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right)\right) \tag{6}$$

To further simplify the problem, we assume that the number of samples in each cluster to be same. Thus, each cluster has $\overline{n} = \frac{N}{K}$ samples. In other words, ${\mathbf{H}^{(m)}}^T \mathbf{H}^{(m)} = \overline{n}\mathbf{I}$. The **Eq**. (6) becomes:

$$\min_{\mathbf{H}^{(m)}} Tr({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}) \tag{7}$$

By performing K-means on each view, we obtain the multi-view K-means as follows:

$$\min_{H^{(m)}} \sum_{m=1}^{M} Tr\left({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right) \tag{8}$$

Moreover, by introducing client weight to each client to measure its contribution to the global model, the objective function becomes:

$$\begin{aligned} &\min_{H^{(m)}} {\alpha^{(m)}}^r \sum_{m=1}^{M} Tr\left({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right) \\ &s.t. \qquad \sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0 \end{aligned} \tag{9}$$

Tensor-based regularizer: As $\mathbf{H}^{(m)}$ is the indicator matrix of the data on m-th client, it may vary across different clients. Since one sample should be assigned to the same cluster in all clients regardless of data heterogeneity, the divergence between all $\mathbf{H}^{(m)}$ should be minimized. Therefore, we introduce a regularizer based on the tensor Schatten p-norm to enforce the model to learn a complementary clustering result, and the objective function becomes:

$$\begin{aligned} &\min_{H^{(m)}} \sum_{m=1}^{M} {\alpha^{(m)}}^r Tr\left({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right) + \lambda ||\boldsymbol{\mathcal{H}}||_{S_p}^p \\ &s.t. \qquad \sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0 \end{aligned} \tag{10}$$

where $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{N \times K \times M}$ is constructed by stacking all $\mathbf{H^{(m)}}$ and λ is the trade-off parameter.

Construction of $\mathbf{D}^{(m)}$: Traditional K-means adopts the Euclidean distance to construct the distance matrix $\mathbf{D}^{(m)}$, which results in high computational complexity and cannot well process linearly non-separable data. Thus, we replace it with an anchor-based affinity matrix [Lu *et al.*, 2023; Li *et al.*, 2020b] and construct $\mathbf{D}^{(m)}$ locally as follows:
Firstly, we utilize directly alternate sampling(DAS) [Li *et al.*, 2020b] to choose the anchors $\mathbf{A}^{(m)} \in \mathbb{R}^{\theta \times d^{(m)}}$ effectively, and θ is the number of anchors.

Secondly, we use square Euclidean distance to measure the difference between samples and anchors. Specifically, according to the parameter-free and effective bipartite graph construct strategy [Li *et al.*, 2020b], we obtain local anchor graph $\mathbf{B}^{(m)}$ according to by solving :

$$\min_{\mathbf{b}_i^{(m)}\mathbf{1}=1, \mathbf{b} \geq \mathbf{0}} \sum_{j=1}^{\theta} h\left(\mathbf{x}_i^{(m)}, \mathbf{a}_j^{(m)}\right) b_{ij}^{(m)} + \gamma \sum_{j=1}^{\theta} \left(b_{ij}^{(m)}\right)^2 \tag{11}$$

where $h\left(\mathbf{x}_i^{(m)}, \mathbf{a}_j^{(m)}\right)$ is the square Euclidean distance of i-th sample and j-th anchor of m-th view, and $\mathbf{b}_i^{(m)}$ is the i-th row of $\mathbf{B}^{(m)}$. Moreover, the regularization parameter γ can be adaptively determined. The anchor points can well cover the entire samples, thus alternately sampling data points according to their feature representations that could reflect some similarity relations over the clusters.

Thirdly, by introducing the Lagrangian function to the closed form of **Eq**. (11), we have:

$$\begin{aligned} \boldsymbol{\mathcal{L}}\left(\mathbf{b}_i^{(m)}, \eta, \mu\right) =& \frac{1}{2} ||\mathbf{b}_i^{(m)} + \frac{\mathbf{p}^{i(m)}}{2\gamma}||_2^2 \\ &+ \eta(\mathbf{b}_i^{(m)}\mathbf{1} - 1) + \mathbf{b}_i^{(m)}\mu \end{aligned} \tag{12}$$

where η and μ are the Lagrangian multipliers and $p_{ij}^{(m)} = h\left(\mathbf{x}_i^{(m)}, \mathbf{a}_j^{(m)}\right)$. According to KKT conditions, we have:

$$\begin{cases} \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \mathbf{b}_i^{(m)}} = b_{ij}^{(m)} + \frac{p_{ij}^{(m)}}{2\gamma} - \eta - \mu_j = 0, \ \ j = 1, ..., \theta \\ b_{ij}^{(m)} \geq 0, \ \ j = 1, ..., \theta \\ \mu_j \geq 0, \ \ j = 1, ..., \theta \\ b_{ij}^{(m)} \mu_j = 0 \end{cases} \tag{13}$$

Fourthly, Supposing each sample is only connected to its k adaptive anchors and solving **Eq**. (13), we have $\mu = \frac{1}{k} + \frac{1}{2\gamma k} \sum_{j=1}^{k} p_{ij}^{(m)}$. Following [Li *et al.*, 2020b], we set $\gamma = \frac{k}{2} p_{i,k+1}^{(m)} - \frac{1}{2} \sum_{j=1}^{k} p_{ij}^{(m)}$, and we can obtain $b_{ij}^{(m)}$ by:

$$b_{ij}^{(m)} = \left(\frac{h_{i,k+1} - h_{i,j}}{k h_{i,k+1} - \sum_{j=1} k h_{i,j}}\right)_+ \tag{14}$$

where $(a)_+ = \max(0, a)$.

Fifthly, the symmetric and doubly-stochastic adjacency matrix $\mathbf{W}^{(m)}$ is obtained according to [Liu *et al.*, 2010]:

$$\mathbf{W}(m) = \mathbf{B}^{(m)} {\Delta^{(m)}}^{-1} {\mathbf{B}^{(m)}}^T \tag{15}$$

where $\Delta_{jj}^{(m)} = \sum_{i=1}^{N} b_{ij}^{(m)}$ and it is a diagonal matrix.

Finally, based on $\mathbf{W}^{(m)}$, we obtained the adjacency matrix $\mathbf{D}^{(m)}$ by:

$$d_{ij}^{(m)} = \sqrt{\frac{1}{1 + \left(\frac{w_{ij}^{(m)}}{\Omega}\right)^4}} \tag{16}$$

where Ω is a hyperparameter. By leveraging the adjacency matrix instead of the Euclidean distance matrix, our method can well handle linearly non-separatable data and can also reduce the computational complexity [Lu *et al.*, 2023; Li *et al.*, 2020b].

3.3 Optimization Algorithm

In federated scenarios, considering data security, $\mathbf{D}^{(m)}$ obtained from local data $\mathbf{X}^{(m)}$ cannot be transmitted to the server or other clients. However, $\mathbf{H}^{(m)}$, which indicates the cluster assignment based on local data, does not cause data leakage. Therefore, it can be transmitted between the server and the client. Based on the above constraints, we propose the following optimization algorithm. Firstly, we introduce augmented Lagrange multiplier(ALM) to **Eq**. (10):

$$\begin{aligned}\mathcal{L}\left(\mathcal{H}, \mathcal{G}\right) = &\sum_{m=1}^{M} {\alpha^{(m)}}^{r} Tr\left({\mathbf{H}^{(m)}}^{T}\mathbf{D}^{(m)}\mathbf{H}^{(m)}\right) \\ &+ \lambda||\mathcal{G}||_{S_p}^{p} + \langle \mathcal{Q}, \mathcal{H} - \mathcal{G}\rangle + \frac{\mu}{2}||\mathcal{H} - \mathcal{G}||_F^2 \quad (17) \\ s.t. \quad &\sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0\end{aligned}$$

where $\mathcal{Q}$ is the Lagrange multiplier and μ is the penalty parameter. Through observation of **Eq**. (17), we can find that:

- There are three variables that need to be optimized, namely $\alpha^{(m)}$, $\mathcal{H}$, and $\mathcal{G}$.
- $\alpha^{(m)}$ can be updated locally on $\mathcal{C}_m$ because it is only related to the local variables $\mathbf{H}^{(m)}$ and $\mathbf{D}^{(m)}$;
- The update of $\mathcal{H}$ seems to involve all the data. To allow updating it locally, we decompose it into M independent subproblems that are only related to local variables $\alpha^{(m)}$ and variables $\mathcal{G}$ and $\mathcal{Q}$ transmitted from the server.
- The update of $\mathcal{G}$ involves t-SVD operation, and thus it needs to be performed globally. Moreover, the update of $\mathcal{G}$ relates to $\mathcal{Q}$ and $\mathcal{H}$.

The federated training of the three variables requires several communication rounds between clients and the server. The data that needs to be transmitted include:(1) each client transmits its updated $\mathbf{H}^{(m)}$ to the server, and (2) the server transmits $\mathbf{Q}^{(m)}$ and $\mathbf{G}^{(m)}$ to the m-th client after $\mathcal{Q}$ and $\mathcal{G}$ are updated. The detailed optimization algorithm is described as follows.

(1)Solving $\alpha^{(m)}$ with fixed $\mathcal{H}$ and $\mathcal{G}$ by $\mathcal{C}_m$: When $\mathcal{H}$ and $\mathcal{G}$ are fixed, we only need to focus on the first term:

$$\begin{aligned}\min_{\alpha^{(m)}} &\sum_{m=1}^{M} {\alpha^{(m)}}^{r} Tr\left({\mathbf{H}^{(m)}}^{T}\mathbf{D}^{(m)}\mathbf{H}^{(m)}\right) \\ s.t. \quad &\sum_{m=1}^{M} \alpha^{(m)} = 1, \alpha^{(m)} \geq 0\end{aligned} \tag{18}$$

By introducing the Lagrange multiplier to **Eq**. (18), we get:

$$\begin{aligned}\mathcal{L}\left(\gamma, \alpha_1, \cdots, \alpha_M\right) = &\sum_{m=1}^{M} {\alpha^{(m)}}^{r} Tr\left({\mathbf{H}^{(m)}}^{T}\mathbf{D}^{(m)}\mathbf{H}^{(m)}\right) \\ &+ \gamma\left(1 - \sum_{m=1}^{M}\alpha^{(m)}\right) - \sum_{m=1}^{M}\theta_m\alpha^{(m)}\end{aligned} \tag{19}$$

where γ and θ_m are hyperparameters. According to KKT conditions, we have the following equations satisfied for all $m = 1, 2, \cdots, M$:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \alpha^{(m)}} = 0 \\ \theta_m \alpha^{(m)} = 0 \\ \theta_m \geq 0 \\ 1 - \sum_{m=1}^{M} \alpha^{(m)} = 0 \end{cases} \tag{20}$$

where $\frac{\partial \mathcal{L}}{\partial \alpha^{(m)}} = r{\alpha^{(m)}}^{r-1} Tr\left({\mathbf{H}^{(m)}}^{T}\mathbf{D}^{(m)}\mathbf{H}^{(m)}\right) - \gamma - \theta_m$. By solving this, we can get the optimization function of $\alpha^{(m)}$:

$$\alpha^{(m)} = \frac{Tr\left({\mathbf{H}^{(m)}}^{T}\mathbf{D}^{(m)}\mathbf{H}^{(m)}\right)^{\frac{1}{1-r}}}{\sum_{m=1}^{M} Tr\left({\mathbf{H}^{(m)}}^{T}\mathbf{D}^{(m)}\mathbf{H}^{(m)}\right)^{\frac{1}{1-r}}} \tag{21}$$

(2)Solving $\mathcal{H}$ with fixed $\alpha^{(m)}$ and $\mathcal{G}$ by $\mathcal{S}$: With fixed $\alpha^{(m)}$ and $\mathcal{G}$, the objective function w.r.t $\mathcal{H}$ becomes:

$$\begin{aligned}\min_{\mathcal{H}} &\sum_{m=1}^{M} {\alpha^{(m)}}^{r} Tr\left({\mathbf{H}^{(m)}}^{T}\mathbf{D}^{(m)}\mathbf{H}^{(m)}\right) \\ &+ \langle \mathcal{Q}, \mathcal{H} - \mathcal{G}\rangle + \frac{\mu}{2}||\mathcal{H} - \mathcal{G}||_F^2\end{aligned} \tag{22}$$

Eq. (22) can be changed into:

$$\begin{aligned}\min_{\mathcal{H}} &\sum_{m=1}^{M} {\alpha^{(m)}}^{r} Tr\left({\mathbf{H}^{(m)}}^{T}\mathbf{D}^{(m)}\mathbf{H}^{(m)}\right) \\ &+ \frac{\mu}{2}||\mathcal{H} - \mathcal{G} + \frac{\mathcal{Q}}{\mu}||_F^2\end{aligned} \tag{23}$$

Apparently, the first term is only related to $\mathbf{H}^{(m)}$ which is the cluster assignment on m-th client, and the second term involves $\mathbf{H}^{(m)}$ of all the clients. Considering the independence of $\mathcal{H}$, $\mathcal{G}$, and $\mathcal{Q}$, we decompose **Eq**. (23) into M independent subproblems, where the m-th subproblem is defined as below:

$$\begin{aligned}\min_{\mathbf{H}^{(m)}} &{\alpha^{(m)}}^{r} Tr\left({\mathbf{H}^{(m)}}^{T}\mathbf{D}^{(m)}\mathbf{H}^{(m)}\right) \\ &+ \frac{\mu}{2}||\mathbf{H}^{(m)} - \mathbf{G}^{(m)} + \frac{\mathbf{Q}^{(m)}}{\mu}||_F^2\end{aligned} \tag{24}$$

Furthermore, we can decompose **Eq**. (24) into N subproblems at the sample level because of the independence of all rows of $\mathbf{H}^{(m)}$.
Regarding the first item of **Eq**. (24), we have:

$$\begin{aligned}&\min_{\mathbf{H}^{(m)}} {\alpha^{(m)}}^r Tr\left({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right)\\ =&\min_{\mathbf{H}^{(m)}} {\alpha^{(m)}}^r \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij} \sum_{k=1}^{c}(h_{ik}^{(m)} h_{jk}^{(m)})\end{aligned} \tag{25}$$

When solving the $\mathbf{h}_t^{(m)}$, fix all $\mathbf{h}_i^{(m)} (i \neq t)$, and considering the duality of i and j and $d_{ii}^{(m)} = 0$, then the function becomes:

$$\min_{\mathbf{h}_t^{(m)}} 2{\alpha^{(m)}}^r {\mathbf{h}_t^{(m)}}^T {\mathbf{H}^{(m)}}^T d_t^{(m)} \tag{26}$$

Then, **regarding the second term of Eq. (24)** , we have:

$$\min_{\mathbf{H}^{(m)}} \frac{\mu}{2} ||\mathbf{H}^{(m)} - \mathbf{G}^{(m)} + \frac{\mathbf{Q}^{(m)}}{\mu}||_F^2 \tag{27}$$

Similarly to solving the first term, we can obtain:

$$\begin{aligned}&\min_{h_t^{(m)}} \frac{\mu}{2} ||\mathbf{h}_t^{(m)} - \mathbf{g}_t^{(m)} + \frac{\mathbf{q}_t^{(m)}}{\mu}||_F^2\\ \Leftrightarrow &\min_{h_t^{(m)}} -\mu {\mathbf{h}_t^{(m)}}^T \mathbf{g}_t^{(m)} + {\mathbf{h}_t^{(m)}}^T \mathbf{q}_t^{(m)}\end{aligned} \tag{28}$$

where $\mathbf{G}^{(m)} = [\mathbf{g}_1^{(m)}, \mathbf{g}_2^{(m)}, \cdot\ \cdot\ \cdot, \mathbf{g}_n^{(m)}]^T$, $\mathbf{Q}^{(m)} = [\mathbf{q}_1^{(m)}, \mathbf{q}_2^{(m)}, \cdots, \mathbf{q}_n^{(m)}]^T$. With the above analyses, we have:

$$\begin{aligned}&\min_{\mathbf{H}^{(m)}} {\alpha^{(m)}}^r Tr\left({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right) + \frac{\mu}{2} ||\mathbf{H}^{(m)} - \mathbf{Z}^{(m)}||_F^2\\ \Leftrightarrow &\min_{\mathbf{h}_t^{(m)}} {\mathbf{h}_t^{(m)}}^T \left(2{\alpha^{(m)}}^r {\mathbf{H}^{(m)}}^T \mathbf{d}_t^{(m)} - \mu \mathbf{z}_t^{(m)}\right)\end{aligned} \tag{29}$$

where $\mathbf{Z}_t^{(m)} = \mathbf{G}^{(m)} - \frac{\mathbf{Q}^{(m)}}{\mu} = [\mathbf{z}_1^{(m)}, \mathbf{z}_2^{(m)}, \cdots, \mathbf{z}_n^{(m)}]^T$. As $\mathbf{h}_t^{(m)}$ is a one-hot vector, the optimization of **Eq**. (29) is:

$$h_{tj}^{(m)} = \begin{cases} 1, j = \arg\min\limits_p \left(2{\alpha^{(m)}}^r {\mathbf{H}^{(m)}}^T \mathbf{d}_t^{(m)} - \mu \mathbf{z}_t^{(m)}\right)_p \\ 0, \text{otherwise}\end{cases} \tag{30}$$

(3)Solving $\mathcal{G}$ with fixed $\alpha^{(m)}$ and $\mathcal{H}$: With fixed $\alpha^{(m)}$ and $\mathcal{H}$, the objective function w.r.t $\mathcal{G}$ becomes:

$$\begin{aligned}&\min_{\mathcal{G}} \lambda ||\mathcal{G}||_{S_p}^p + \langle \mathcal{Q}, \mathcal{H} - \mathcal{G}\rangle + \frac{\mu}{2}||\mathcal{H} - \mathcal{G}||_F^2\\ =&\min_{\mathcal{G}} \lambda ||\mathcal{G}||_{S_p}^p + \frac{\mu}{2}||\mathcal{G} - \mathcal{P}||_F^2\end{aligned} \tag{31}$$

where $\mathcal{P} = \mathcal{H} - \frac{\mathcal{Q}}{\mu}$. From ***Definition 4***, we can get the optimal solution of **Eq**. (31):

$$\mathcal{G}^* = \text{ifft}\left(\mathcal{U} * \mathcal{D}_{\frac{\lambda}{\mu}, p}(\mathcal{P}) * \mathcal{V}^T\right) \tag{32}$$

where $\mathcal{U}$ and $\mathcal{V}$ are acquired via t-SVD of $\mathcal{P}$, *i.e.,* $\mathcal{P} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$.

Algorithm 1 TensorFMVC

Input: The data $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(M)}\}$ on M local clients, pre-constructed $\mathbf{D}^{(m)}$, cluster number C;
Parameter: μ, p, λ, r, Ω;
Output: Cluster assignment $\mathbf{H}$

Initialize $\mathbf{H}^{(m)}$ with random one-hot vector and $\alpha^{(m)}$=$\frac{1}{M}$ for all client. Initialize $\mathcal{Q}$ and $\mathcal{G}$ to be all zero.
while not converged **do**
  for $m = 1\ to\ M$ **do**
    ▷ **on** m**-th client** $\mathcal{C}_m$
    Update weight $\alpha^{(m)}$ according to **Eq**. (21)
    Update $\mathbf{H}^{(m)}$ according to **Eq**. (30)
    Send $\mathbf{H}^{(m)}$ and $\alpha^{(m)}$ to the global server
    Compute $Tr\left({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right)^{\frac{1}{1-r}}$ and send it to the global server
  end for
  ▷ **on Server** $\mathcal{S}$
  Stack all $\mathbf{H}^{(m)}$ to construct tensor $\mathcal{H}$
  Update $\mathcal{G}$ with new $\mathcal{H}$ according to **Eq**. (32)
  Update $\mathcal{Q}$ according to $\mathcal{Q} \leftarrow \mathcal{Q} + \mu(\mathcal{H} - \mathcal{G})$
  Send $\mathbf{H}^{(m)}$, $\mathbf{G}^{(m)}$, and $\mathbf{Q}^{(m)}$ to $\mathcal{C}_m$
  Compute $\sum_{m=1}^{M} Tr\left({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right)^{\frac{1}{1-r}}$ and send it to all clients
end while
$\mathcal{S}$ aggregates $\mathbf{H}^{(m)}$ into $\mathbf{H}$ with **Eq**. (33)
return $\mathbf{H}$

Workflow: We depict the workflow in **Fig**. 1 and summarize it in ***Alg.*** 1 to illustrate how parameters are optimized collaboratively. Firstly, each client $\mathcal{C}_m$ conducts initialization by setting $\mathbf{H}^{(m)}$ with random one-hot vectors and $\alpha^{(m)}$ with $\frac{1}{M}$, and $\mathcal{Q}$ and $\mathcal{G}$ as zero tensors. After initialization, $\mathcal{C}_m$ updates $\mathbf{H}^{(m)}$with **Eq**. (30), and computes $Tr\left({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right)^{\frac{1}{1-r}}$, then sends them and $\alpha^{(m)}$ to the global server $\mathcal{S}$. The server $\mathcal{S}$, after receiving all the parameters, stacks all the $\mathbf{H}^{(m)}$ into a tensor $\mathcal{H}$, and then update $\mathcal{G}$ with **Eq**. (32) and $\mathcal{Q}$ with $\mathcal{Q} + \mu(\mathcal{H} - \mathcal{G})$. Subsequently, $\mathcal{S}$ will also compute $\sum_{m=1}^{M} Tr\left({\mathbf{H}^{(m)}}^T \mathbf{D}^{(m)} \mathbf{H}^{(m)}\right)^{\frac{1}{1-r}}$ and send it together with updated $\mathcal{Q}$ and $\mathcal{G}$ to all the clients, with which the client $\mathcal{C}_m$ could update $\alpha^{(m)}$ with **Eq**. (21). The above process (except the initialization) repeats until convergence. Finally, the server $\mathcal{S}$ aggregates the latest $\mathbf{H}^{(m)}$ into the final clustering indicator matrix $\mathbf{H}$:

$$h_{ij} = \begin{cases} 1 & j = \arg\min\limits_p \left(\sum_{m=1}^{M} {\alpha^{(m)}}^r \mathbf{h}_i^{(m)}\right)_p \\ 0 & \text{otherwise}\end{cases} \tag{33}$$

Complexity Analysis: We evaluate computational complexity: For client $\mathcal{C}_m$, the complexity of pre-constructing $\mathbf{D}^{(m)}$ and model update is $\mathcal{O}\left(MN\theta d^{(m)} + MN\theta log(\theta)\right)$ and $\mathcal{O}\left(NC\right)$. For the server $\mathcal{S}$, the complexity of aggregation is $\mathcal{O}\left(MNClog(MN) + M^2NC\right)$. Both of them are linear to N indicating its low computational complexity.

Datasets	3-sources			BBCSport			ORL			Sonar		
Metrics	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
DiMSC	69.23	63.13	74.56	85.85	70.62	85.85	77.75	90.09	80.50	55.77	1.25	55.77
MvLRSSC	55.92	49.81	70.59	62.87	40.47	64.63	63.51	80.01	66.86	50.48	3.12	53.37
RMSL	31.95	14.46	41.42	76.63	72.36	76.63	83.00	93.16	87.75	62.02	4.29	62.02
GMC	70.74	65.15	79.29	80.33	**73.89**	84.01	42.25	68.37	52.75	50.48	4.50	53.37
MvDGNMF	30.77	21.11	33.14	82.54	67.32	82.54	65.50	79.52	69.50	62.50	4.66	62.50
UDBGL	34.91	5.60	35.50	36.40	2.43	36.58	59.25	77.36	62.50	57.21	1.61	57.21
FastMICE	55.62	50.25	71.01	43.93	11.16	45.40	78.75	90.46	82.25	58.17	3.23	58.17
FedMVL	60.36	50.12	67.46	65.07	49.23	73.90	51.75	66.84	55.00	64.90	7.04	64.90
TensorFMVC	**72.19**	**66.18**	**81.07**	**86.94**	68.71	**86.94**	**99.75**	**99.77**	**99.75**	**98.56**	**89.16**	**98.56**

Table 1: Clustering performance comparison in terms of ACC, NMI, and PUR on 3-sources, BBCSport, ORL, and Sonar datasets.

Datasets	Yale			VS			HAR			RGB-D		
Metrics	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
DiMSC	44.85	52.84	44.85	68.95	22.29	68.95	51.79	32.14	25.69	39.61	32.67	49.76
MvLRSSC	44.06	48.02	45.09	56.78	6.12	56.78	49.38	53.56	53.40	39.00	32.40	50.59
RMSL	78.78	78.23	**79.39**	67.50	11.90	67.50	48.64	52.99	55.38	12.63	2.85	26.98
GMC	21.21	27.51	24.24	80.43	28.75	80.43	48.04	57.40	48.60	40.23	33.06	46.51
MvDGNMF	36.36	42.70	38.79	50.06	0.60	50.06	46.36	35.21	46.36	26.50	0.78	27.26
UDBGL	52.73	65.94	54.55	51.26	0.05	51.26	47.78	46.20	50.45	43.89	35.96	53.55
FastMICE	62.42	57.01	65.46	51.49	0.09	51.69	56.79	49.58	56.79	41.81	32.61	49.53
FedMVL	49.70	54.12	50.91	76.35	22.58	76.35	53.68	54.70	43.71	33.26	45.55	22.32
TensorFMVC	**79.39**	**78.48**	**79.39**	**99.94**	**99.30**	**99.94**	**70.67**	**61.80**	**70.67**	**57.76**	**63.83**	**80.12**

Table 2: Clustering performance comparison in terms of ACC(%), NMI(%), and PUR(%) on Yale, VS, HAR, and RGB-D datasets.

4 Experiment

4.1 Experiment Settings

We validate our method on eight multi-view datasets and compare it with seven multi-view clustering methods and one federated multi-view clustering method. For the federating scenario, our experiment includes a server and multiple clients, and each client holds the data with one view.

Datasets: We evaluate our method on eight public multi-view datasets: (1)**3-sources** is a three-view text dataset with 169 samples sourced from three reputable news outlets. (2)**BBCSport** [Greene and Cunningham, 2006] is composed of 544 sports news articles sourced from the BBC Sport website and is categorized into five distinct topical areas. It has two views and the dimensions are 3283 and 3183. (3)**ORL** [Samaria and Harter, 1994] is a three-view dataset of 400 facial images, categorized into 40 classes. (4)**Sonar** [Sejnowski and Gorman,] includes three views and extracts its multi-view features from 208 patterns. (5)**Yale** is a two-view dataset of 165 facial images of 11 people. (6)**Vehicle Sensor((VS)** [Duarte and Hu, 2004] is a four-view dataset whose features are gathered from distributed sensors. (7)**Human Activity Recognition(HAR)** [Reyes-Ortiz and Parra, 2012] is a four-view dataset with 10299 samples that documents six daily activities performed; (8)**SentencesNYU v2(RGB-D)** [Silberman *et al.*, 2012] includes images of indoor scenes and corresponding descriptions. We process the dataset following the method in [Trosten *et al.*, 2021].

Compared methods: We compared our methods with seven multi-view clustering methods: (1)**DiMSC** [Cao *et al.*, 2015];(2)**MvLRSSC** [Brbić and Kopriva, 2018];(3)**RMSL** [Li *et al.*, 2019];(4)**GMC** [Wang *et al.*, 2019];(5)**MvDGNMF** [Li *et al.*, 2020a];(6)**UDBGL**[Fang *et al.*, 2023a];(7)**FastMICE**[Huang *et al.*, 2023]; and one federated method (8)**FedMVL** [Huang *et al.*, 2022];

4.2 Experiment Results

The experimental results are illustrated in **Table** 1 and **Table** 2. The best results are in **bold**, and the second best results are underlined. We can reach the following conclusions: (1) Our method shows superiority to the compared federated method(*i.e.,* FedMVL). Compared with the FedMVL on the ORL dataset, the proposed method is 48.00%, 32.93%, and 44.75% higher on ACC, NMI, and PUR, demonstrating TensorFMVC effectively overcomes the weaknesses of existing federated multi-view clustering methods. (2) TensorFMVC achieves the best performance even when compared with centralized methods. For example, compared with the best-performance centralized method on ORL, TensorFMVC is 16.75%, 6.61%, and 12.00% higher on ACC, NMI, and PUR. This may be because the tensor Schatten p-norm explores the complementary information between various views.

Convergence analysis: **Fig.** 2 shows the convergence curves of ACC and the values of the objective function. It shows that within limited iterations, both the ACC and **Eq**.

variant	*case1*			*case2*			*case3*			TensorFMVC		
metrics	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
ORL	77.25	87.31	80.00	92.25	94.09	92.25	63.25	80.14	65.50	99.75	99.77	99.75
Sonar	52.88	0.22	53.37	76.44	21.11	76.44	62.50	4.79	62.50	98.56	89.16	98.56
Yale	61.21	61.78	62.42	46.67	51.97	46.67	36.97	43.87	38.97	79. 39	78.48	79.39
VS	95.86	75.17	95.86	90.56	55.15	90.56	70.77	17.15	70.77	99.94	99.30	99.94

Table 3: Results of ablation studies on ORL, Sonar, Yale, VS.

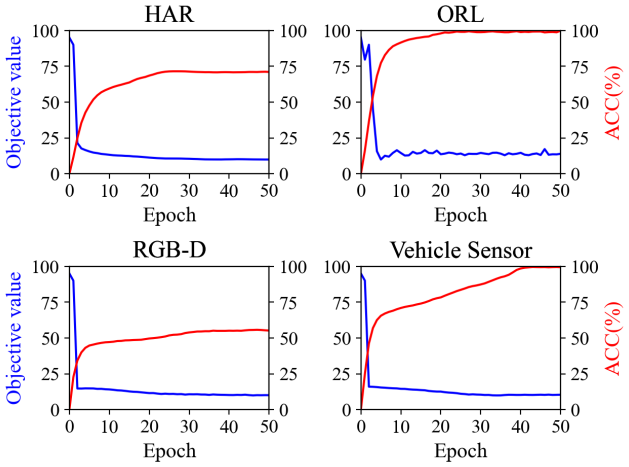

Figure 2: Convergence curves on HAR, ORL, RGB-D, and VS.

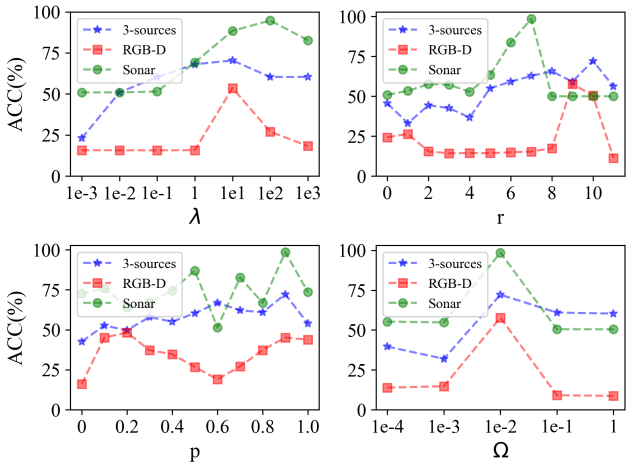

Figure 3: Parameter analysis on 3-sources, RGB-D, and Sonar.

(10) reach the convergence state rapidly. The convergence curve can be divided into three parts. In the first stage, the value of the objective function decreases rapidly and ACC increases rapidly. In the second stage, the value of the objective function decreases relatively slowly while the accuracy rate increases slowly as well. Eventually, the two tend to converge. Moreover, the proposed method possesses a stable convergence because the objective function values can reach a stable value within 50 iterations on the four datasets.

Parameter analysis: There are three parameters in **Eq**. (10), *i.e.,* λ, p, and r, and one parameter Ω in **Eq**. (16). We study the influence of their values on the clustering performance, whose results are illustrated in **Fig.** 3. We conclude that: (1) TensorFMVC achieves relatively poor performance when the λ is too small or too large, and the probable reason is that λ adjusts the contribution of the tensor Schatten p-norm. Thus, a small λ makes the model unable to learn the global cluster assignment correctly, while a big value results in ignoring the complementary information of different values. The suggested range for λ is [10, 100]. (2) A larger r represents the smaller difference in client contributions to the global model. The ACC curve w.r.t. r indicates that an appropriate increase in r helps to improve the clustering accuracy. Thus, the suggested range of r is 5 to 7. (3) p's value greatly influences the ACC, because when $0 \leq p \leq 1$, the tensor Schatten p-norm helps to obtain a better approximation than other norms. It achieves the best performance when $p = 0.9$ because it learns the spatial structure from all clients. (4) The ACC curve w.r.t. Ω illustrates that the clustering performance deteriorates when Ω is too large or too small. Hence, the recommended value of Ω is 0.01.

Ablation Study: We have conducted ablation experiments in three cases: (1) TensorFMVC without tensor-based regularizer(***case 1***);(2) TensorFMVC without anchor-based affinity graph(***case 2***)(By using Euclidean distance instead); (3) TensorFMVC without tensor-based regularizer and anchor-based affinity graph(***case 3***) and the results are shown in **Table.** 3. Results indicate TensorFMVC achieves better performance compared with all three cases, showing the significant contribution of each component.

5 Conclusion

This paper presents a novel federated multi-view clustering method with tensor factorization named TensorFMVC. It effectively addresses the sensitivity of K-means to centroid initialization through centerless K-means. Additionally, it employs an anchor-based affinity matrix instead of Euclidean distance to reduce complexity and improve efficiency. Inspired by tensor-based methods, we introduce the Schatten p-norm to extract the complementary information shared across all views. To validate the proposed method in federated scenarios, we develop a distributed optimization algorithm, enabling the model to be optimized collaboratively. Experimental results demonstrate the efficacy of TensorFMVC and its superiority to state-of-the-art methods.

Acknowledgements

This research was partially supported by the National Key Research and Development Project of China No.2021ZD011070, National Science Foundation of China under Grant No. 62102306, the Natural Science Basic Research Program of Shaanxi Province under Grant No. 2023-JC-YB-534, the Science and Technology project of Xian under Grant No. 2022JH-JSYF-0009, Initiative Postdocs Supporting Program under Grant No. BX20190262.

References

[Brbić and Kopriva, 2018] Maria Brbić and Ivica Kopriva. Multi-view low-rank sparse subspace clustering. *Pattern Recognition*, 73:247–258, 2018.

[Cai *et al.*, 2013] Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In *Twenty-Third International Joint conference on artificial intelligence*, 2013.

[Cai *et al.*, 2023] Hecheng Cai, Yuze Tan, Shudong Huang, and Jiancheng Lv. Lifelong multi-view spectral clustering. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3488–3496, 2023.

[Cao *et al.*, 2015] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–594, 2015.

[Chen *et al.*, 2020] Chuan Chen, Yu Wang, Weibo Hu, and Zibin Zheng. Robust multi-view k-means clustering with outlier removal. *Knowledge-Based Systems*, 210:106518, 2020.

[Chen *et al.*, 2023] Xinyue Chen, Jie Xu, Yazhou Ren, Xiaorong Pu, Ce Zhu, Xiaofeng Zhu, Zhifeng Hao, and Lifang He. Federated deep multi-view clustering with global self-supervision. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3498–3506, 2023.

[Dong *et al.*, 2022] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10164–10173, 2022.

[Duarte and Hu, 2004] Marco F Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004.

[Fang *et al.*, 2023a] Si-Guo Fang, Dong Huang, Xiao-Sha Cai, Chang-Dong Wang, Chaobo He, and Yong Tang. Efficient multi-view clustering via unified and discrete bipartite graph learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[Fang *et al.*, 2023b] Uno Fang, Man Li, Jianxin Li, Longxiang Gao, Tao Jia, and Yanchun Zhang. A comprehensive survey on multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Feng and Yu, 2020] Siwei Feng and Han Yu. Multi-participant multi-class vertical federated learning. *arXiv preprint arXiv:2001.11154*, 2020.

[Gao *et al.*, 2020] Quanxue Gao, Pu Zhang, Wei Xia, Deyan Xie, Xinbo Gao, and Dacheng Tao. Enhanced tensor rpca and its application. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):2133–2140, 2020.

[Greene and Cunningham, 2006] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384, 2006.

[Han *et al.*, 2020] Junwei Han, Jinglin Xu, Feiping Nie, and Xuelong Li. Multi-view k-means clustering with adaptive sparse memberships and weight allocation. *IEEE Transactions on Knowledge and Data Engineering*, 34(2):816–827, 2020.

[Hu *et al.*, 2023] Xingchen Hu, Jindong Qin, Yinghua Shen, Witold Pedrycz, Xinwang Liu, and Jiyuan Liu. An efficient federated multi-view fuzzy c-means clustering method. *IEEE Transactions on Fuzzy Systems*, 2023.

[Huang *et al.*, 2018] Shudong Huang, Yazhou Ren, and Zenglin Xu. Robust multi-view data clustering with multi-view capped-norm k-means. *Neurocomputing*, 311:197–208, 2018.

[Huang *et al.*, 2022] Shudong Huang, Wei Shi, Zenglin Xu, Ivor W Tsang, and Jiancheng Lv. Efficient federated multi-view learning. *Pattern Recognition*, 131:108817, 2022.

[Huang *et al.*, 2023] Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Kong *et al.*, 2018] Hao Kong, Xingyu Xie, and Zhouchen Lin. t-schatten-p norm for low-rank tensor recovery. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1405–1419, 2018.

[Li *et al.*, 2019] Ruihuang Li, Changqing Zhang, Huazhu Fu, Xi Peng, Tianyi Zhou, and Qinghua Hu. Reciprocal multi-layer subspace learning for multi-view clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8172–8180, 2019.

[Li *et al.*, 2020a] Jianqiang Li, Guoxu Zhou, Yuning Qiu, Yanjiao Wang, Yu Zhang, and Shengli Xie. Deep graph regularized non-negative matrix factorization for multi-view clustering. *Neurocomputing*, 390:108–116, 2020.

[Li *et al.*, 2020b] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):330–344, 2020.

[Li *et al.*, 2023] Xingfeng Li, Zhenwen Ren, Quansen Sun, and Zhi Xu. Auto-weighted tensor schatten p-norm for robust multi-view graph clustering. *Pattern Recognition*, 134:109083, 2023.

[Liu *et al.*, 2010] Wei Liu, Junfeng He, and Shih-Fu Chang. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 679–686, 2010.

[Liu *et al.*, 2019] Yongli Liu, Xiaoqin Zhang, Guiying Tang, and Di Wang. Multi-view subspace clustering based on tensor schatten-p norm. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5048–5055. IEEE, 2019.

[Liu *et al.*, 2023] Chengliang Liu, Jie Wen, Zhihao Wu, Xiaoling Luo, Chao Huang, and Yong Xu. Information recovery-driven deep incomplete multiview clustering network. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[Lu *et al.*, 2023] Han Lu, Quanxue Gao, Qianqian Wang, Ming Yang, and Wei Xia. Centerless multi-view k-means based on the adjacency matrix. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8949–8956, 2023.

[Pei *et al.*, 2022] Shenfei Pei, Huimin Chen, Feiping Nie, Rong Wang, and Xuelong Li. Centerless clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):167–181, 2022.

[Reyes-Ortiz and Parra, 2012] Anguita Davide Ghio Alessandro Oneto Luca Reyes-Ortiz, Jorge and Xavier Parra. Human Activity Recognition Using Smartphones. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C54S4K.

[Samaria and Harter, 1994] Ferdinando S Samaria and Andy C Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE workshop on applications of computer vision*, pages 138–142. IEEE, 1994.

[Sejnowski and Gorman,] Terry Sejnowski and R. Gorman. Connectionist Bench (Sonar, Mines vs. Rocks). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5T01Q.

[Silberman *et al.*, 2012] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012.

[Trosten *et al.*, 2021] Daniel J Trosten, Sigurd Lokse, Robert Jenssen, and Michael Kampffmeyer. Reconsidering representation alignment for multi-view clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1255–1265, 2021.

[Wang *et al.*, 2019] Hao Wang, Yan Yang, and Bing Liu. Gmc: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1116–1129, 2019.

[Wang *et al.*, 2023] Xu Wang, Dezhong Peng, Peng Hu, Yunhong Gong, and Yong Chen. Cross-domain alignment for zero-shot sketch-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 7024–7035, 2023.

[Xing *et al.*, 2023] Lei Xing, Haiquan Zhao, Zhiping Lin, and Badong Chen. Mixture correntropy based robust multi-view k-means clustering. *Knowledge-Based Systems*, 262:110231, 2023.

[Xu *et al.*, 2022] Jie Xu, Yazhou Ren, Huayi Tang, Zhimeng Yang, Lili Pan, Yang Yang, Xiaorong Pu, S Yu Philip, and Lifang He. Self-supervised discriminative feature learning for deep multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Xu *et al.*, 2023] Jie Xu, Yazhou Ren, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. Untie: Clustering analysis with disentanglement in multi-view information fusion. *Information Fusion*, 100:101937, 2023.

[Yan *et al.*, 2023] Weiqing Yan, Yuanyang Zhang, Chenlei Lv, Chang Tang, Guanghui Yue, Liang Liao, and Weisi Lin. Gcfagg: Global and cross-view feature aggregation for multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19863–19872, 2023.

[Yang and Sinaga, 2019] Miin-Shen Yang and Kristina P Sinaga. A feature-reduction multi-view k-means clustering algorithm. *IEEE Access*, 7:114472–114486, 2019.

[Zhang *et al.*, 2019] Xiaoqian Zhang, Huaijiang Sun, Zhigui Liu, Zhenwen Ren, Qiongjie Cui, and Yanmeng Li. Robust low-rank kernel multi-view subspace clustering based on the schatten p-norm and correntropy. *Information Sciences*, 477:430–447, 2019.

[Zhao *et al.*, 2022] Yujiao Zhao, Yu Yun, Xiangdong Zhang, Qin Li, and Quanxue Gao. Multi-view spectral clustering with adaptive graph learning and tensor schatten p-norm. *Neurocomputing*, 468:257–264, 2022.