Large-scale Multi-view Tensor Clustering with Implicit Linear Kernels

Jiyuan Liu[1], Xinwang Liu[1]*, Chuankun Li[2], Xinhang Wan[1], Hao Tan[1], Yi Zhang[1], Weixuan Liang[1]
Qian Qu[1], Yu Feng[1], Renxiang Guan[1], Ke Liang[1]
[1] National University of Defense Technology, Changsha, Hunan, China. 410072.
[2] North University of China, Taiyuan, Shanxi, China. 030051.
liujiyuan13@nudt.edu.cn, ⋆Corresponding author

Abstract

Multi-view clustering is a long-standing hot topic in machine learning communities, due to its capability of integrating data information from multiple sources and modalities. By utilizing tensor Singular Value Decomposition (t-SVD) technique with the tensor rotation trick, recent advances have achieved remarkable improvements on clustering performance. However, we find this is attributed to the inadvertent use of sequential information of sorted data samples, i.e. inadvertent label use, which violates the unsupervised learning setting. On the other hand, existing large-scale approaches are mostly developed on the basis of matrix factorization or anchor techniques, thereby fail to consider the similarities among all data samples, preventing from further performance improvement. To address the above issues, we first analyze the tensor rotation trick and recommend to remove it from tensor clustering. On its basis, a novel large-scale multi-view tensor clustering method is developed by incorporating the pair-wise similarities with implicit linear kernel function. To solve the resultant optimization problem, we design an efficient algorithm of linear complexity. Moreover, extensive experiments are conducted and corresponding results well support the aforementioned finding and validate the effectiveness and efficiency of the proposed method.

1. Introduction

With the rapid development of electrical device and information technology, the data is collected from multiple sources with different modalities, which is widely known as multi-view data. In literature, it refers to the multiple descriptions of a same set of data samples which can be separated semantically. For instance, in the Alzheimer's disease diagnosis, the doctor would probably collect data from a series of medical tests, such as neurological examination, ElectroEncephaloGram (EEG), Computed Tomography (CT) scan, etc. In this background, multi-view clustering is a representative of unsupervised learning paradigms to deal with multi-view data and achieves promising performance in recent years [13, 28–31, 43, 44, 46, 47, 52], since it can enhance the consistent information while integrate the complementary information of different data views. Apart from the rich research interests in academic research, the methods are also applied in numerous real-world scenarios, such as Bioinformatics [24, 51], Geoscience [6], Internet of Things [9], etc.

Recently, a volume of multi-view clustering methods employ tensor techniques to utilize the high-order correlations among different data views, which is annotated to multi-view tensor clustering [5, 18, 19, 50]. In the pioneer research, a tensor nuclear norm minimization constraint is proposed on the basis of tensor-Singular Value Decomposition (t-SVD) and imposed on the clustering objective function, leading to remarkable performance improvements [50]. Also, other constraints are developed based on t-SVD subsequently, such as the popular weighted tensor Schatten p-norm minimization [10, 48, 49], enhanced tensor rank minimization [16], etc. To deal with large-scale data, a volume of multi-view clustering methods aims to design algorithms and corresponding optimization strategies of linear complexity. Some of them are developed on the basis of matrix factorization where the data is decomposed into two parts, i.e. coefficient matrix and base matrix [8, 11, 14, 27]. On contrary, another branch of them reduces the computation complexity by building kernel matrix or similarity matrix with sample anchors of limited numbers [15, 23].

Among the above methods, two drawbacks are observed and severely limit the further development of multi-view clustering. On one hand, almost all multi-view tensor clustering approaches employ the tensor rotation trick, as shown in Fig. 2, along with incorporating the t-SVD based constraints. Since t-SVD requires to adopt Fast Fourier Transform (FFT) on the 3-rd dimension of target tensor, the data sequential information is used inadvertently. Unfortunately, the existing methods overlook this and perform clustering on the data sorted in groups, thereby introducing the data

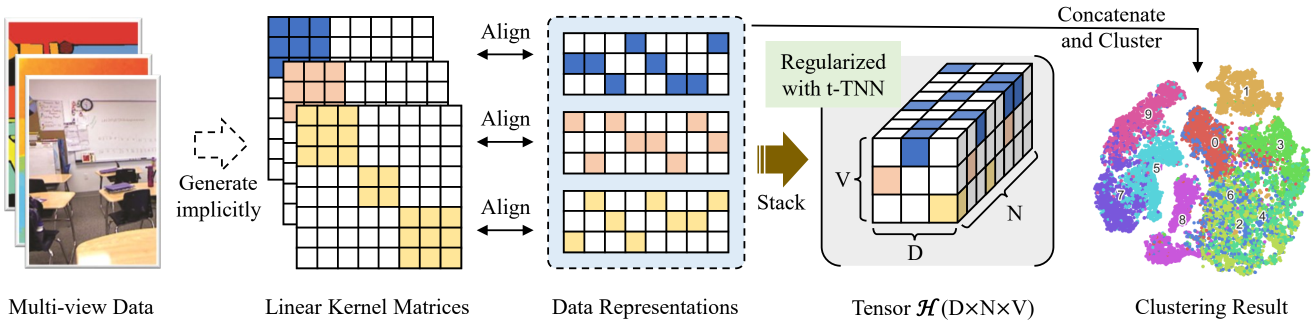

Figure 1. Framework of the proposed large-scale multi-view tensor clustering method. For brevity, only three data views are presented (similar is to any number of data views). Note that, the linear kernel matrices are not required to be computed explicitly. Meanwhile, different from existing approaches, the tensor rotation trick **is not** employed on tensor $\mathcal{H}$.

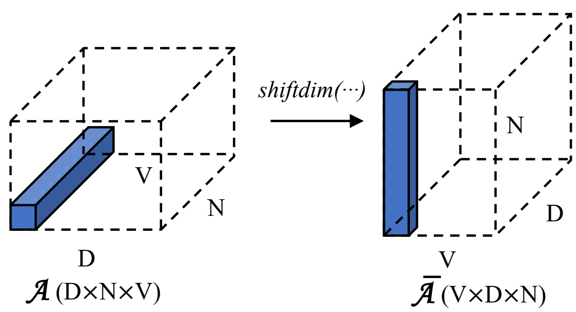

Figure 2. The tensor rotation trick which can be implemented with *shiftdim* function in MATLAB. Note that, N, D and V refer to the sample number, dimension of data representation or anchor number and view number, respectively. When tensor $\mathcal{A}$ is of size $N \times N \times V$ in subspace or spectral clustering framework [45], tensor $\bar{\mathcal{A}}$ is obtained in size $V \times N \times N$.

labels and violating the unsupervised learning setting. On the other hand, existing large-scale approaches are mostly developed on the basis of matrix factorization and anchor techniques, thereby fail to consider the similarities among all data samples, preventing from further performance improvement.

To address the above issues, we first analyze the tensor rotation trick and recommend to remove it from tensor clustering. On this basis, a novel Large-scale Multi-view Tensor Clustering (LMTC) method, as presented in Fig. 1, is developed by incorporating the pair-wise similarity of data samples. In specific, with given multi-view data, corresponding linear kernel matrices can be generated. Note that, since the complexity is square to sample number, they are not required to be computed explicitly. Then, the latent data representations can be formulated by aligning with the implicit kernels respect to each view. Meanwhile, they are supposed to be stacked into a tensor and regularized with the t-SVD based Tensor Nuclear Norm (t-TNN) constraint. Finally, the data can be categorized via concatenating the latent representations and applying classical clustering technique, such as k-means, etc. The above procedures are modeled into a unified objective function. To solve the resultant optimization problem, we design an efficient algorithm of linear complexity. Moreover, extensive experiments are conducted and corresponding results well support the aforementioned finding and validate effectiveness of the proposed LMTC method. Overall, the contributions can be summarized as follows:

1) To the first attempt, we find the tensor rotation trick inadvertently invokes sequential information of sorted data samples, thereby violates the unsupervised learning setting. Since almost all existing multi-view tensor clustering methods do so, the finding will have a great impact.
2) A novel large-scale multi-view tensor clustering method is proposed by removing the tensor rotation trick and incorporating the pair-wise similarity of data samples.
3) We develop an optimization algorithm of linear complexity to solve the proposed method. Also, extensive experiments are conducted to validate its effectiveness and efficiency.

2. Related work

2.1. Multi-view tensor clustering

Multi-view tensor clustering is an unsupervised learning paradigm by employ tensor techniques to utilize the high-order correlations among different data views [3, 5, 18, 19, 50]. In the pioneer research, Xie et al. construct a unified tensor by computing and stacking the view-specific subspace representation matrices [50]. Then, they propose the tensor nuclear norm minimization constraint by adopting the well-known tensor-Singular Value Decomposition (t-SVD) [18, 19], and impose it on the aforementioned tensor, contributing to a novel t-SVD based multi-view subspace clustering method. In experiments, it achieves remarkable performance improvements on benchmarks and rapidly attracts the interests of a large amount of multi-view clustering researchers. Apart from the subspace representation, existing methods also generate the unified tensor with other data representation matrices, such as latent representation [4, 22], anchor graph [16, 49], etc. In addition, a large number of novel t-SVD based constraints are also developed. For example, Gao et al. propose the weighted tensor Schatten p-norm minimization by regularizing the larger singular values shrink less to preserve the prominent information of tensors [10]. Ji et al. formulate the enhanced tensor rank minimization to uncover the local geometric structure in anchor representation tensor [16].

2.2. Large-scale multi-view clustering

In literature, a volume of multi-view clustering approaches are proposed to deal with large-scale data by developing algorithms and optimization strategies of linear complexity. They can be roughly grouped into two categories. The first category originates from the matrix factorization technique in which the data is decomposed into two parts, i.e. coefficient matrix and base matrix [8, 11, 14, 27]. For example, Liu et al. compute a set of coefficient matrices with each corresponding to one data view and push them towards a common consensus for further clustering task [8]. Gao et al. assume that all data views share a consensus manifold and only adopt one unique coefficient matrix to capture the underlying clustering structure [11]. Moreover, the other category proposes to reduce computation complexity by building anchor graphs with selecting a limited number of data samples, such as [7, 15, 23, 36]. Nevertheless, tensor techniques are also utilized in them [16, 22, 37, 41, 49]. For instance, Li et al. propose the orthogonal non-negative tensor factorization to exploit the within-view spatial structure and between-view complementary information [22]. Xia et al. propose a variance-based de-correlation anchor selection strategy for bipartite construction and employ the tensor Schatten p-norm to exploit the inter-view similarity of data samples [49].

3. Notation and preliminary

Before the proposed method, the necessary notations are introduced in the following. Generally, the bold calligraphy letter, e.g. $\mathcal{A}$, is used for tensor, the bold upper case letter, e.g. $\mathbf{A}$, is used for matrix, the bold lower case letter, e.g. $\mathbf{a}$, is used for vector, while the normal letter and the aforementioned letters with subscript index are used for element, such as a, $\mathcal{A}_{i,j,\dots}$, $\mathbf{A}_{i,j}$ and $\mathbf{a}_i$. As well, the fiber of a tensor refers to a 2-dimension section by fixing all but two indices, while the slice of a tensor to a 1-dimension section by fixing all but one index. Given a 3-way tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, its mode-1, mode-2 and mode-3 fibers are denoted to $\mathcal{A}(:, i, j)$, $\mathcal{A}(i, :, j)$ and $\mathcal{A}(i, j, :)$. At the same time, its i-th horizontal, lateral and frontal slices are denoted to $\mathcal{A}(i, :, :)$, $\mathcal{A}(:, i, :)$ and $\mathcal{A}(:, :, i)$. For ease of expression, $\mathbf{A}^{(i)}$ is used to represent $\mathcal{A}(:, :, i)$. Also, five more tensor operations in [19] are used, including bcirc, bvec, bvfold, bdiag and bdfold, and can be found in Appendix. Moreover, to introduce t-SVD and t-TNN, the following definitions should be clarified in advance.

Algorithm 1 Computation of t-SVD [19]

Input: Tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$
Output: Tensor $\mathcal{U}$, $\mathcal{S}$, $\mathcal{V}$

1: $\mathcal{A}_f = \mathrm{fft}(\mathcal{A}, 3)$;
2: **for** $m = 1 : N_3$ **do**
3: $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \mathrm{SVD}(\mathcal{A}_f^{(m)})$;
4: $\mathcal{U}_f^{(m)} = \mathbf{U}$, $\mathcal{S}_f^{(m)} = \mathbf{S}$, $\mathcal{V}_f^{(m)} = \mathbf{V}$;
5: **end for**
6: $\mathcal{U} = \mathrm{ifft}(\mathcal{U}_f, 3)$, $\mathcal{S} = \mathrm{ifft}(\mathcal{S}_f, 3)$, $\mathcal{V} = \mathrm{ifft}(\mathcal{V}_f, 3)$;

Definition 1 (t-product) *With given two arbitrary tensors $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$, their t-product is defined to*

$$\mathcal{A} * \mathcal{B} = \mathrm{fold}(\mathrm{bcirc}(\mathcal{A}) \cdot \mathrm{bvec}(\mathcal{B})) \tag{1}$$

whose size is $N_1 \times N_4 \times N_3$.

Definition 2 (Orthogonal Tensor) *A tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is orthogonal if*

$$\mathcal{A}^\top * \mathcal{A} = \mathcal{A} * \mathcal{A}^\top = \mathcal{I}, \tag{2}$$

where $\mathcal{A}^\top$ is the transposed tensor of size $N_2 \times N_1 \times N_3$ and obtained by transposing the frontal slices of $\mathcal{A}$ then reversing the order of transposed frontal slices 2 through n. Meanwhile, $\mathcal{I}$ is the identity tensor of size $N_2 \times N_2 \times N_3$ whose first frontal slice is the identity matrix and the other frontal slices are all zeros.

Definition 3 (f-diagonal Tensor) *A tensor is f-diagonal if each of its frontal slices is diagonal.*

Definition 4 (tensor Singular Value Decomposition) *Given an arbitrary tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, the tensor Singular Value Decomposition (t-SVD) is given as*

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top, \tag{3}$$

where $\mathcal{U}$ and $\mathcal{V}$ are orthogonal tensors of size $N_1 \times N_1 \times N_3$ and $N_2 \times N_2 \times N_3$, while $\mathcal{S}$ is f-diagonal tensor of size $N_1 \times N_2 \times N_3$ and $*$ refers to the tensor t-product.

With respect to Definition 3, Kilmer et al. also explore its correlations to Singular Value Decomposition (SVD) in Fourier domain and propose to compute it efficiently via Algorithm 1. Note that, $\mathcal{A}_f = \mathrm{fft}(\mathcal{A}, 3)$ refers to the fast Fourier transform of tensor $\mathcal{A}$ along the 3-rd dimension, while $\mathcal{A} = \mathrm{ifft}(\mathcal{A}_f, 3)$ is the inverse operation.

Definition 5 (Tensor Nuclear Norm) *Given an arbitrary tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, the t-SVD based Tensor Nuclear Norm (t-TNN) is given as*

$$\|\mathcal{A}\|_{\circledast} = \sum_{i=1}^{\min(N_1,N_2)} \sum_{k=1}^{N_3} |\mathcal{S}_f(i,i,k)|. \tag{4}$$

According to [19], t-TNN in Definition 5 is the tightest convex relaxation to ℓ_1-norm of the tensor multi-rank, thereby achieves better performances than other norms in machine learning tasks, such as tensor completion [26], etc.

4. Methodology

4.1. Inadvertent label use

As introduced in Section 2.1, existing multi-view tensor clustering methods first construct the target tensor by stacking the latent data representations [4, 22], self-representation matrices [45], etc. Then, the tensor rotation trick, as presented in Fig. 2, is applied on it along with t-SVD based constraints to integrate the data clustering information of different views. For ease of expression, with annotating the tensor and the trick to be $\mathcal{A}$ and $shiftdim$ function[1], the rotated tensor can be computed by

$$\bar{\mathcal{A}} = shiftdim(\mathcal{A}), \tag{5}$$

where $\mathcal{A}$ is mostly of size $D \times N \times V$ and $N \times N \times V$, while $\bar{\mathcal{A}}$ of size $V \times D \times N$ and $V \times N \times N$. Note that, the 3-rd dimension of tensor $\bar{\mathcal{A}}$ is always the number of data samples.

According to Definiton 4, 5 and Alg. 1, it requires applying FFT on tensor $\bar{\mathcal{A}}$ along the 3-rd dimension to compute t-TNN $\|\bar{\mathcal{A}}\|_{\circledast}$. Strictly, this can be written in mode-3 fiber, i.e.

$$\bar{\mathcal{A}}_f(i,j,:) = \text{fft}(\bar{\mathcal{A}}(i,j,:)). \tag{6}$$

where $\bar{\mathcal{A}}_f(i,j,:)$ and $\bar{\mathcal{A}}(i,j,:)$ are both of size $N \times 1$. Nevertheless, it is worth to note that fiber $\bar{\mathcal{A}}(i,j,:)$ is composed of N elements (similarity, feature, etc.) with each corresponding to to a unique data sample. Since FFT is to project a value sequence from the time domain to the frequency domain [38], $\bar{\mathcal{A}}_f(i,j,:)$ encodes not only the value information but also the sequential information of $\bar{\mathcal{A}}(i,j,:)$.

In multi-view tensor clustering literature, almost all approaches directly use the public benchmarks, such as ORL (please refer to Section 5.1), which are sorted by groups in advance, but do not shuffle the data before feeding into their clustering algorithms [16, 22, 33, 49]. Although remarkable performances are achieved, they inadvertently invoke the data sequential information (also annotated to inadvertent label use) in Eq. (6), hence violate the unsupervised learning setting. This can be validated by comparing their performances on shuffled data with those on sorted data and we provide corresponding experiment results in Section 4.1.

To address the aforementioned inadvertent label use problem, there are two possible solutions: 1) removing the tensor rotation trick; 2) adopting the tensor rotation trick but shuffling the data in advance. Taking tensor $\mathcal{A}$ and $\bar{\mathcal{A}}$ in Eq. (5) for example, the former only needs $\mathcal{O}((V \log V)DN)$ or $\mathcal{O}((V \log V)N^2)$ complexity to compute the t-TNN, while the latter requires $\mathcal{O}(VDN \log N)$ or $\mathcal{O}(VN^2 \log N)$ complexity which is at least $\log N$ times higher than the former. Therefore, we recommend to not use the tensor rotation trick in future tensor clustering researches.

4.2. Objective function

In large-scale multi-view clustering literature, existing approaches are mostly developed on the basis of matrix factorization or anchor techniques, thereby fail to consider the similarities among all data samples, preventing from further performance improvement. To address this issue, we propose the LMTC method by incorporating the pair-wise similarity with linear kernel function.

Given multi-view data $\{\mathbf{X}_v\}_{v=1}^V$ of K clusters in which $\mathbf{X}_v \in \mathbb{R}^{D_v \times N}$ (V, N and D_v are the number of data views, the number of samples and the dimension of v-th data view, respectively), the linear kernel matrix of the v-th data view can be written as

$$\mathbf{K}_v = \mathbf{X}_v^\top \mathbf{X}_v, \tag{7}$$

in which $\mathbf{K}_v \in \mathbb{R}^{N \times N}$ and its (i,j)-th element is the pair-wise similarity between the i-th sample and the j-th sample. Assuming each data view corresponds to a low-rank manifold, we can denote the latent representation of v-th view to $\mathbf{H}_v$. Also, the dimensions of all latent representations are universally set to $K \times N$. By following the research in [27], each row of representation $\mathbf{H}_v$ are expected to be discriminative from the others, which is formulated by seting the latent representations orthogonal, i.e.

$$\mathbf{H}_v \mathbf{H}_v^\top = \mathbf{I}_K, \tag{8}$$

where $\mathbf{I}_K$ refers to the identity matrix of size K. Nevertheless, the latent representation $\mathbf{H}_v$ should be aligned with the linear kernel matrices $\mathbf{K}_v$ of Eq. (7), which can be achieved by maximizing the following

$$\max_{\mathbf{H}_v} \frac{< \mathbf{K}_v, \mathbf{H}_v^\top \mathbf{H}_v >}{\|\mathbf{K}_v\|_F \cdot \|\mathbf{H}_v^\top \mathbf{H}_v\|_F}. \tag{9}$$

Considering the orthogonal property of $\mathbf{H}_v$ in Eq. (8), $\|\mathbf{H}_v^\top \mathbf{H}_v\|_F$ is a constant and Eq. (9) can be transformed into

$$\max_{\mathbf{H}_v} \text{Tr}(\mathbf{K}_v \mathbf{H}_v^\top \mathbf{H}_v). \tag{10}$$

Furthermore, we also stack all latent representations into a unified tensor $\mathcal{H} \in \mathbb{R}^{K \times N \times V}$ and maximize its t-TNN to

[1]The tensor rotation trick is implemented with *shiftdim* function in MATLAB. Here we use the symbol directly.

capture the low-rank clustering structure underlying all data views, i.e.

$$\min_{\mathcal{H}} \|\mathcal{H}\|_{\circledast}, \tag{11}$$

where $\mathcal{H} = \mathrm{bvfold}([\mathbf{H}_1; \mathbf{H}_2; \cdots; \mathbf{H}_V])$. Combining Eq. (8), (10) and (11), we can obtain the following

$$\begin{aligned} \min_{\{\mathbf{H}_v\}_{v=1}^V} & -\sum_{v=1}^{V} \mathrm{Tr}(\mathbf{K}_v \mathbf{H}_v^\top \mathbf{H}_v) + \lambda \|\mathcal{H}\|_{\circledast} \\ & s.t.\ \mathbf{H}_v \mathbf{H}_v^\top = \mathbf{I}_K, \end{aligned} \tag{12}$$

where λ is the trade-off parameter and should be set in advance. Since the matrix $\mathbf{K}_v$ requires the $\mathcal{O}(N^2)$ computation and storage complexity, we do not calculate it explicitly and get the objective function as

$$\begin{aligned} \min_{\{\mathbf{H}_v\}_{v=1}^V} & -\sum_{v=1}^{V} \mathrm{Tr}(\mathbf{X}_v^\top \mathbf{X}_v \mathbf{H}_v^\top \mathbf{H}_v) + \lambda \|\mathcal{H}\|_{\circledast} \\ & s.t.\ \mathbf{H}_v \mathbf{H}_v^\top = \mathbf{I}_K. \end{aligned} \tag{13}$$

By optimizing Eq. (13), we can be obtain the feasible latent representations of all data views. Then, the cluster assignment is computed by applying the classical k-means on their vertical concatenation.

4.3. Optimization

To solve the objective function of Eq. (13), we employ the Augmented Lagrange Multiplier (ALM) [25] and design an alternate optimization algorithm of linear complexity. By introducing an auxiliary tensor variable $\mathcal{G} \in \mathbb{R}^{K \times N \times V}$, corresponding unconstrained optimization problem can be obtained as

$$\begin{aligned} L(\{\mathbf{H}_v\}_{v=1}^V, \mathcal{G}, \mathcal{W}) = & -\sum_{v=1}^{V} \mathrm{Tr}(\mathbf{X}_v^\top \mathbf{X}_v \mathbf{H}_v^\top \mathbf{H}_v) \\ & + \lambda \|\mathcal{G}\|_{\circledast} + < \mathcal{W},\ \mathcal{G} - \mathcal{H} > + \frac{\rho}{2} \|\mathcal{G} - \mathcal{H}\|_F^2 \\ & s.t.\ \mathbf{H}_v \mathbf{H}_v^\top = \mathbf{I}_K, \end{aligned} \tag{14}$$

where tensor $\mathcal{W}$ is the Lagrange multiplier and ρ represents the penalty parameter adjusted along with optimization. On this basis, the alternate optimization algorithm can be illustrated by the following three subproblems.

$\mathbf{H}_v$-subproblem. With fixing the latent representations $\{\mathbf{H}_{v'}\}_{v'=1, v' \neq v}^V$, tensor $\mathcal{G}$ and Lagrange multiplier $\mathcal{W}$, the problem of Eq. (14) can be reduced to

$$\begin{aligned} & \max_{\mathbf{H}_v} \mathrm{Tr}(\mathbf{H}_v \mathbf{X}_v^\top \mathbf{X}_v \mathbf{H}_v^\top) + \mathrm{Tr}(\mathbf{H}_v^\top \mathbf{C}) \\ & s.t.\ \mathbf{H}_v \mathbf{H}_v^\top = \mathbf{I}_K, \end{aligned} \tag{15}$$

in which

$$\mathbf{C} = \rho \mathbf{G}^{(v)} + \mathbf{W}^{(v)}. \tag{16}$$

Algorithm 2 Optimization of $\mathbf{H}_v$-subproblem

Input: data $\mathbf{X}_v$, matrix $\mathbf{C}$ and initial representation $\mathbf{H}_v$
Output: representation $\mathbf{H}_v$

ensure $\mathbf{H}_v \mathbf{H}_v^\top = \mathbf{I}_K$;
repeat
  compute $\mathbf{M} = \mathbf{X}_v^\top \mathbf{X}_v \mathbf{H}_v^\top + \mathbf{C}$;
  perform SVD on $\mathbf{M}$, i.e. $\mathbf{U} \Sigma \mathbf{V}^\top = \mathbf{M}$;
  update $\mathbf{H}_v = \mathbf{V} \mathbf{U}^\top$;
until convergent

Algorithm 3 Optimization algorithm of LMTC method

Input: multi-view data $\{\mathbf{X}_v\}_{v=1}^V$ and cluster number K
Output: cluster assignment

initialize $\{\mathbf{H}_v\}_{v=1}^V$ such that $\mathbf{H}_v \mathbf{H}_v^\top = \mathbf{I}_D$, $\mathcal{G} = 0$, $\mathcal{W} = 0$, $\rho = 10^{-5}$, $\rho_{\max} = 10^{10}$, $\eta = 2$, $\delta_1 = 10^{-5}$, $\delta_2 = 10^{-7}$ and $t = 1$;
while not convergent **do**
  for $v \in \{1, 2, \cdots, V\}$ **do**
    update latent representation $\mathbf{H}_v$ with Alg. 2;
  end for
  update tensor $\mathcal{G}$ with Eq. (18);
  update Lagrange multiplier $\mathcal{W}$ with Eq. (21)
  compute the objective value obj^t with Eq. (13);
  check the convergence conditions:
  $obj^t - obj^{t-1} / obj^t \leq \delta_1$ and
  $\|\mathcal{G} - \mathcal{H}\|_F \leq \delta_2$
  update parameter $\rho = \min(\eta\rho, \rho_{\max})$;
  $t = t + 1$;
end while
compute cluster assignment by applying k-means on the concatenation of latent representations $\{\mathbf{H}_v\}_{v=1}^V$;

It is easy to find that Eq. (15) is a quadratic optimization problem on the Stiefel manifold [35] and can be solved by Alg. 2.

$\mathcal{G}$-subproblem. With fixing the latent representations $\{\mathbf{H}_v\}_{v=1}^V$, tensor $\mathcal{G}$ and Lagrange multiplier $\mathcal{W}$, the problem of Eq. (14) can be formulated into

$$\min_{\mathcal{G}} \frac{\lambda}{\rho} \|\mathcal{G}\|_{\circledast} + \frac{1}{2} \|\mathcal{G} - \mathcal{P}\|_F^2. \tag{17}$$

where $\mathcal{P} = \mathcal{H} - \mathcal{W}/\rho$. By following Theorem 2 of [50], we can obtain the solution that

$$\mathcal{G} = \mathrm{ifft}(\mathcal{G}_f,\ 3) \tag{18}$$

with v-th frontal slice of $\mathcal{G}_f^{(v)}$ being

$$\mathcal{G}_f^{(v)} = \mathcal{U}_f^{(v)} \mathcal{C}_{\rho,\lambda}(\mathcal{S}_f^{(v)}) \mathcal{V}_f^{(v)\top}, \tag{19}$$

where

$$\begin{aligned}&\mathcal{U}_f^{(v)}\mathcal{S}_f^{(v)}\mathcal{V}_f^{(v)\top}=\mathcal{P}_f^{(v)}\\&\mathcal{C}_{\rho,\lambda}(\mathcal{S}_f^{(v)})=\max\{0,\ \mathcal{S}_f^{(v)}-\lambda V/\rho\}.\end{aligned}\tag{20}$$

$\mathcal{W}$**-subproblem**. By fixing the latent representations $\{\mathbf{H}_v\}_{v=1}^V$, tensor $\mathcal{H}$ and tensor $\mathcal{G}$, the Lagrange multiplier $\mathcal{W}$ can be updated via

$$\mathcal{W}=\mathcal{W}+\rho(\mathcal{G}-\mathcal{H}).\tag{21}$$

Overall, the complete optimization algorithm of LMTC method is summarized in Alg. 3.

4.4. Computation complexity

In the following, computation complexity of the proposed LMTC method is analyzed by each subproblem.

1) $\mathbf{H}_v$**-subproblem**. In the computation of matrix $\mathbf{M}$, $\mathbf{X}_v^\top\mathbf{X}_v\mathbf{H}_v^\top$ can be efficiently calculated as $\mathbf{X}_v^\top(\mathbf{X}_v\mathbf{H}_v^\top)$, hence the $\mathcal{O}(KD_vN)$ complexity is required. Also, the SVD on matrix M and the update of $\mathbf{H}_v$ both needs $\mathcal{O}(K^2N)$ complexity.
2) $\mathcal{G}$**-subproblem**. The tensor $\mathcal{P}$ is transformed to $\mathcal{P}_f$ by applying FFT along the 3-rd dimension, leading to $\mathcal{O}((V\log V)KN)$ complexity. So does the inverse FFT from tensor $\mathcal{G}_f$ to $\mathcal{G}$. Meanwhile, the SVD on matrix $\mathcal{P}_f^{(v)}$ and the computation of matrix $\mathcal{G}_f^{(v)}$ is of $\mathcal{O}(K^2N)$ complexity.
3) $\mathcal{W}$**-subproblem**. The computation of Eq. (21) only consists of tensor additions of size $K\times N\times V$, resulting in the $\mathcal{O}(VKN)$ complexity.

To be summarized, assuming t iterations are required to converge, the optimization algorithm of LMTC method is of $\mathcal{O}(tN)$ computation complexity, which is linear to the number of data samples.

5. Experiment

5.1. Setting

In the following experiments, we test the proposed LMTC method on seven popular benchmark datasets, including ORL [39], HW [42], BDGP [40], ALOI [12], DryBean [20], AwA [21] and YtVideo [34], whose specifics can be found in Table 1. Meanwhile, the proposed LMTC method is also compared with ten classic and novel large-scale multi-view clustering approaches, including RMKC [2], BMVC [53], LMSC [17], OPMC [27], EOMSC [32], MCHBG [54], ASR-ETR [16], S^{2}MVTC [33], TBGL [49] and Orth-NTF [22]. To ensure comparison fairness, the aforementioned comparative approaches are all of **linear complexity** to the number of data samples and the latter four are specifically chosen from researches in last three years which also leverage the t-SVD based tensor techniques to improve the clustering performance. Nevertheless, we directly run their codes publicly available at the authors' websites without further revision for reproduction. Note that, **the data samples are shuffled to remove the effect of inadvertent label use by default**. In addition, we grid-search the parameters recommended in corresponding papers and report the best. So do the proposed LMTC method with setting the parameter λ in $[10^{-10},10^{-9},\cdots 10^{5}]$. By following the multi-view clustering literature, three widely-used metrics, i.e. Accuracy (ACC), Normalized Mutual Information (NMI) and Purity, are adopted to evaluate the clustering results. Furthermore, the methods are executed multiple times to remove randomness and their averages are presented. Additionally, in the following tables, *Error* indicates corresponding algorithm terminates with errors, while *OT* (*Out of Time*) and *OM* (*Out of Memory*) refers to consequences of excedding the maximal time limit and computing memory, respectively. By the way, the code of LMTC is released on https://github.com/liujiyuan13/LMTC-code_release.

Table 1. Details of the used benchmark datasets.

Dataset	Number of Samples	Views	Clusters
ORL	400	3	40
HW	2000	6	10
BDGP	2500	3	5
ALOI	10800	4	100
DryBean	13611	2	7
AwA	30475	6	50
YtVideo	101499	5	31

5.2. Performance comparison

To validate effectiveness of the proposed LMTC method, we test it on seven benchmarks and compare the results with those of the completing ones in Table 2. It can be observed that the proposed LMTC method requires to set 1 parameter before computing the clustering results. In comparison, BMVC, LMSC, EOMSC, MCHBG, ASR-ETR, S^{2}MVTC, TBGL and Orth-NTF are supposed to set 6, 2, 2, 3, 4, 3, 5 and 3 parameters respectively. Since multi-view clustering is an unsupervised learning paradigm which lacks of supervisory signals to tune parameters and parameters are mostly set manually based on experience, a satisfying performance is hard to guarantee with more parameters. Although OPMC is free of parameter and RMKC only has 1 parameter, their clustering performances are far worse than those of the proposed LMTC. Besides, OPMC, compared with LMTC, takes nearly $10\times$ to $20\times$ time to group the data into categories, as shown in Table 3.

As for clustering performance comparison, the pro-

Table 2. Performance comparison between the proposed LMTC method and completing approaches in literature. Note that, *Param.* is short for *Parameter number*. In addition, the best and second-best results are marked in bold and with underline, respectively.

Dataset	RMKC	BMVC	LMSC	OPMC	EOMSC	MCHBG	ASR-ETR	S^2MVTC	TBGL	Orth-NTF	LMTC
Param.	1	6	2	**0**	2	3	4	3	5	3	1
						ACC					
ORL	57.35	57.25	59.00	56.00	62.25	66.25	82.75	*Error*	57.05	39.70	**82.97**
HW	70.43	86.40	92.10	87.51	76.00	85.40	87.15	57.15	75.64	66.32	**94.00**
BDGP	49.63	32.00	45.00	50.69	42.08	28.12	50.08	31.56	*OT*	*Error*	**54.81**
ALOI	43.12	53.80	65.56	51.47	23.76	60.67	57.98	55.84	*OT*	40.77	**69.07**
DryBean	55.24	50.45	70.60	47.63	60.23	68.25	72.07	49.14	*OT*	28.52	**74.80**
AwA	9.01	10.45	8.18	9.40	8.72	*OM*	9.41	6.62	*OT*	4.45	**10.65**
YtVideo	12.38	19.41	17.25	18.29	**26.66**	*OM*	17.05	11.25	*OT*	*OT*	25.31
						NMI					
ORL	75.53	72.19	78.91	74.58	88.15	75.08	90.98	*Error*	70.15	57.40	**91.60**
HW	70.71	84.03	86.49	81.19	82.08	**88.08**	76.75	64.40	70.39	58.15	87.12
BDGP	25.69	8.20	24.57	**35.37**	14.59	3.52	23.85	7.44	*OT*	*Error*	32.84
ALOI	47.21	53.87	77.34	69.73	58.26	72.90	76.08	67.86	*OT*	51.83	**79.79**
DryBean	47.16	37.30	57.00	40.33	53.23	59.68	61.20	38.16	*OT*	8.58	**63.19**
AwA	11.16	12.30	9.03	11.95	10.10	*OM*	10.91	5.88	*OT*	2.93	**12.31**
YtVideo	10.17	15.80	14.08	17.68	0.24	*OM*	1.65	8.84	*OT*	*OT*	**22.26**
						Purity					
ORL	61.20	60.00	65.50	60.75	80.12	70.25	84.50	*Error*	63.45	43.15	**85.05**
HW	73.99	86.40	92.10	87.51	76.20	87.65	87.15	63.85	76.00	68.15	**94.00**
BDGP	50.57	33.88	45.96	53.27	42.08	28.20	50.08	33.40	*OT*	*Error*	**56.73**
ALOI	49.56	52.45	69.01	53.41	24.82	63.89	61.13	56.92	*OT*	43.00	**71.11**
DryBean	59.56	57.17	72.00	56.66	61.65	68.78	74.23	56.26	*OT*	34.02	**77.68**
AwA	11.06	12.19	10.03	11.49	9.62	*OM*	11.55	7.29	*OT*	5.99	**12.45**
YtVideo	26.87	30.78	32.25	30.04	26.68	*OM*	26.64	27.01	*OT*	*OT*	**35.07**

Table 3. Time consumptions of the proposed LMTC method and completing approaches in literature. Note that, the time is in seconds.

Dataset	RMKC	BMVC	LMSC	OPMC	EOMSC	MCHBG	ASR-ETR	S^2MVTC	TBGL	Orth-NTF	LMTC
ORL	5.72	0.55	9.18	173.38	8.56	3.85	36.52	*Error*	26.48	2.66	10.55
HW	11.02	3.81	13.37	22.74	1.44	37.91	6.68	6.88	1616.79	13.65	2.91
BDGP	1.10	1.61	5.62	95.27	2.05	30.59	15.58	2.35	*OT*	*Error*	3.90
ALOI	8914.13	17.50	67.80	395.30	42.98	1955.45	187.73	7.36	*OT*	2559.52	353.87
DryBean	2.57	3.24	22.24	79.74	40.51	5530.91	10.20	8.94	*OT*	162.51	5.18
AwA	6830.57	68.90	2379.84	48190.69	326.52	*OM*	2460.72	82.49	*OT*	11724.26	2464.91
YtVideo	59.69	151.24	1422.19	39035.14	766.77	*OM*	2091.84	182.19	*OT*	*OT*	3383.65

posed LMTC method outperforms the completing ones and achieves the best in almost all settings. Concretely, it improves the accuracy respect to the second-best by 0.22%, 1.90%, 4.12%, 3.51%, 2.73% and 0.20% on the former six datasets. Although 1.35% accuracy decrease is observed on YtVideo, corresponding EOMSC method takes two hyperparameters and the associated NMI is completely infeasible. Also, OPMC outperforms the proposed LMTC method by 2.53% NMI on BDGP, but it takes 95.27 seconds which is nearly 24 times to LMTC. By comparing with the latter four multi-view tensor clustering methods in depth, the proposed LMTC outperforms them by large margins. To be summarized, the proposed LMTC method achieves promising performances and validated to be effective empirically.

5.3. Time consumption

To validate the feasibility on large-scale data, we investigate the time consumptions of the proposed LMTC and completing methods on chosen benchmark (ORL) in Table 3. To ensure comparison fairness, each method is allocated to

Table 4. Validation (**ACC**) on inadvertent label use by applying the tensor rotation trick. Note that, the arrows $\downarrow$ and $\uparrow$ represent performance decrease and increase, respectively. Moreover, only ACC results are provided, while NMI and Purity results in Appendix.

Dataset	ASR-ETR			S^2MVTC			TBGL			Orth-NTF		
	Sort	Shuffle	Gap	Sort	Shuffle	Gap	Sort	Shuffle	Gap	Sort	Shuffle	Gap
ORL	88.50	82.75	5.75 $\downarrow$	*Error*	*Error*	*Error*	70.75	57.05	13.70 $\downarrow$	66.25	39.70	26.55 $\downarrow$
HW	99.85	87.15	12.70 $\downarrow$	88.50	57.15	31.35 $\downarrow$	74.55	75.64	1.09 $\uparrow$	82.55	66.32	16.23 $\downarrow$
BDGP	99.04	50.08	48.96 $\downarrow$	99.44	31.56	67.88 $\downarrow$	*OT*	*OT*	*OT*	*Error*	*Error*	*Error*
ALOI	80.03	57.98	22.05 $\downarrow$	55.16	55.84	0.69 $\uparrow$	*OT*	*OT*	*OT*	63.61	40.77	22.84 $\downarrow$
DryBean	91.17	72.07	19.10 $\downarrow$	78.75	49.14	29.61 $\downarrow$	*OT*	*OT*	*OT*	69.41	28.52	40.89 $\downarrow$
AwA	67.70	9.41	58.30 $\downarrow$	56.03	6.62	49.41 $\downarrow$	*OT*	*OT*	*OT*	64.63	4.45	60.18 $\downarrow$
YtVideo	71.35	17.05	54.30 $\downarrow$	54.59	11.25	43.34 $\downarrow$	*OT*	*OT*	*OT*	*OT*	*OT*	*OT*

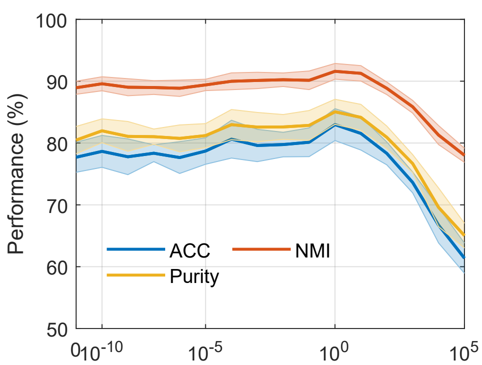

Figure 3. Performance variation respect to parameter λ.

a single CPU core of the same server and repeated multiple times to obtain the averages. It can be observed that LMTC is about the middle among all multi-view clustering methods. Due that the completing methods are of linear complexity to the number of samples and proposed to solve large-scale problem, the LMTC is also feasible to deal with large-scale data. In the latter four multi-view tensor clustering methods, TBGL cannot compute the clustering results on BDGP, ALOI, DryBean, AwA and YtVideo datasets before the maximal time limit. Nevertheless, Orth-NTF takes much longer time than LMTC. As a comparison, these two observations further indicate the efficiency of LMTC.

5.4. Study on the tensor rotation trick

As introduced in Section 4.1, the application of tensor rotation trick along with t-SVD based constraints incorporates the data sequential information and results in inadvertent label use. In existing researches, most of multi-view tensor clustering methods do so and the completing methods, i.e. ASR-ETR, S^2MVTC, TBGL and Orth-NTF, are the representatives. To support this claim, we conduct an ablation study by comparing them in two settings where the first is shuffling data before clustering, while the other is sorting data before clustering. Corresponding results are collected in Table 4. It can obvious that performances of the four methods drop to a large extent when shuffling data, well illustrating the fact that data labels are inadvertently used in existing multi-view tensor clustering approaches.

5.5. Parameter study

Since the proposed LMTC method consists of a trade-off parameter λ, the parameter study is conducted on the chosen benchmark datasets. Specifically, with grid-searching λ in $[10^{-10}, 10^{-9}, \cdots 10^{5}]$, we record the performances and present its variation in Fig. 3. It can be observed that the performance first increases then reaches the top and finally decrease when increasing parameter λ. Since parameter λ can be considered as the weight of item $\|\mathcal{H}\|_{\circledast}$ in Eq. (13), this well validates the effectiveness of the t-TNN constraint to improve clustering performance. Nevertheless, this also suggests us to set λ around 10^{0} in practice.

6. Conclusion

Although existing multi-view clustering methods have achieved remarkable performance improvements in recent years, some of them inadvertently incorporate the data labels by utilizing t-SVD technique with the tensor rotation trick, violating the unsupervised learning setting. Meanwhile, existing large-scale approaches fail to consider the similarities among all data samples. To address the two issues, we propose to remove the tensor rotation trick and, on this basis, develop a novel large-scale multi-view tensor clustering method of linear complexity. Moreover, extensive experiments are conducted and corresponding results well validate its effectiveness and efficiency.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 62306324, 62376279, U24A20333, 62325604, 62276271, 62421002), Science and Technology Innovation Program of Hunan Province (No. 2024RC3128) and National University of Defense Technology Research Foundation (No. ZK24-30).

References

[1] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20(4):1956–1982, 2010. 1

[2] Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2598–2604. IJCAI/AAAI, 2013. 6, 2

[3] Jie Chen, Zhu Wang, Hua Mao, and Xi Peng. Low-rank tensor learning for incomplete multiview clustering. *IEEE Trans. Knowl. Data Eng.*, 35(11):11556–11569, 2023. 2

[4] Man-Sheng Chen, Jia-Qi Lin, Xiang-Long Li, Bao-Yu Liu, Chang-Dong Wang, Dong Huang, and Jian-Huang Lai. Representation learning in multi-view clustering: A literature review. *Data Sci. Eng.*, 7(3):225–241, 2022. 3, 4

[5] Man-Sheng Chen, Chang-Dong Wang, and Jian-Huang Lai. Low-rank tensor based proximity learning for multi-view clustering. *IEEE Trans. Knowl. Data Eng.*, 35(5):5076–5090, 2023. 1, 2

[6] Zhengyi Chen, Chunmin Zhang, Tingkui Mu, and Yifan He. Tensorial multiview subspace clustering for polarimetric hyperspectral images. *IEEE Trans. Geosci. Remote. Sens.*, 60: 1–13, 2022. 1

[7] Zhe Chen, Xiaojun Wu, Tianyang Xu, and Josef Kittler. Fast self-guided multi-view subspace clustering. *IEEE Trans. Image Process.*, 32:6514–6525, 2023. 3

[8] Jing Gao, Jiawei Han, Jialu Liu, and Chi Wang. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA*, pages 252–260. SIAM, 2013. 1, 3

[9] Jing Gao, Meng Liu, Peng Li, Asif Ali Laghari, Abdul Rehman Javed, Nancy Victor, and Thippa Reddy Gadekallu. Deep incomplete multiview clustering via information bottleneck for pattern mining of data in extreme-environment iot. *IEEE Internet of Things Journal*, 11(16): 26700–26712, 2024. 1

[10] Quanxue Gao, Pu Zhang, Wei Xia, De-Yan Xie, Xinbo Gao, and Dacheng Tao. Enhanced tensor RPCA and its application. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(6):2133–2140, 2021. 1, 3

[11] Shengxiang Gao, Zhengtao Yu, Taisong Jin, and Ming Yin. Multi-view low-rank matrix factorization using multiple manifold regularization. *Neurocomputing*, 335:143–152, 2019. 1, 3

[12] Jan-Mark Geusebroek, Gertjan J. Burghouts, and Arnold W. M. Smeulders. The amsterdam library of object images. *Int. J. Comput. Vis.*, 61(1):103–112, 2005. 6, 2

[13] Zongbo Han, Changqing Zhang, Huazhu Fu, and Joey Tianyi Zhou. Trusted multi-view classification with dynamic evidential fusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45 (2):2551–2566, 2023. 1

[14] Shudong Huang, Zenglin Xu, Zhao Kang, and Yazhou Ren. Regularized nonnegative matrix factorization with adaptive local structure learning. *Neurocomputing*, 382:196–209, 2020. 1, 3

[15] Shudong Huang, Zenglin Xu, Ivor W. Tsang, and Zhao Kang. Auto-weighted multi-view co-clustering with bipartite graphs. *Inf. Sci.*, 512:18–30, 2020. 1, 3

[16] Jintian Ji and Songhe Feng. Anchor structure regularization induced multi-view subspace clustering via enhanced tensor rank minimization. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 19286–19295. IEEE, 2023. 1, 3, 4, 6

[17] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. Large-scale multi-view subspace clustering in linear time. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 4412–4419. AAAI Press, 2020. 6, 3

[18] Misha Elena Kilmer and Carla D. Moravitz Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435:641–658, 2011. 1, 2

[19] Misha Elena Kilmer, Karen S. Braman, Ning Hao, and Randy C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM J. Matrix Anal. Appl.*, 34(1): 148–172, 2013. 1, 2, 3, 4

[20] Murat Koklu and Ilker Ali Özkan. Multiclass classification of dry beans using computer vision and machine learning techniques. *Comput. Electron. Agric.*, 174:105507, 2020. 6, 2

[21] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 951–958. IEEE Computer Society, 2009. 6, 2

[22] Jing Li, Quanxue Gao, Qianqian Wang, Ming Yang, and Wei Xia. Orthogonal non-negative tensor factorization based multi-view clustering. In *Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 3, 4, 6

[23] Xingfeng Li, Yuangang Pan, Yuan Sun, Yinghui Sun, Quansen Sun, Zhenwen Ren, and Ivor W. Tsang. Scalable unpaired multi-view clustering with bipartite graph matching. *Information Fusion*, page 102786, 2024. 1, 3

[24] Yunfan Li, Dan Zhang, Mouxing Yang, Dezhong Peng, Jun Yu, Yu Liu, Jiancheng Lv, Lu Chen, and Xiaocui Peng. scbridge embraces cell heterogeneity in single-cell rna-seq and atac-seq data integration. *Nature Communications*, 14, 2023. 1

[25] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Journal of structural biology*, 181 2:116–27, 2010. 5

[26] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):208–220, 2013. 4

[27] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Li Liu, Siqi Wang, Weixuan Liang, and Jiangyong Shi. One-pass multi-view clustering for large-scale data. In *IEEE/CVF International*

Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, pages 12324–12333. IEEE, 2021. 1, 3, 4, 6

[28] Jiyuan Liu, Xinwang Liu, Jian Xiong, Qing Liao, Sihang Zhou, Siwei Wang, and Yuexiang Yang. Optimal neighborhood multiple kernel clustering with adaptive local kernels. *IEEE Trans. Knowl. Data Eng.*, 34(6):2872–2885, 2022. 1

[29] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Xifeng Guo, Marius Kloft, and Liangzhong He. Multiview subspace clustering via co-training robust data representation. *IEEE Trans. Neural Networks Learn. Syst.*, 33(10):5177–5189, 2022.

[30] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Qing Liao, and Yuanqing Xia. Contrastive multi-view kernel learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(8):9552–9566, 2023.

[31] Jiyuan Liu, Xinwang Liu, Zhiping Cai, En Zhu, and Rusheng Ju. On the correlation measurement of data representations. *Chinese Journal of Computers*, 47:1568–1581, 2024. 1

[32] Suyuan Liu, Siwei Wang, Pei Zhang, Kai Xu, Xinwang Liu, Changwang Zhang, and Feng Gao. Efficient one-pass multiview subspace clustering with consensus anchors. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual Event, February 22 - March 1, 2022*, pages 7576–7584. AAAI Press, 2022. 6, 3

[33] Zhen Long, Qiyuan Wang, Yazhou Ren, Yipeng Liu, and Ce Zhu. S^2mvtc: A simple yet efficient scalable multi-view tensor clustering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 24088–24097. IEEE, 2024. 4, 6, 3

[34] Omid Madani, Manfred Georg, and David A. Ross. On using nearly-independent feature families for high precision and confidence. *Mach. Learn.*, 92(2-3):457–477, 2013. 6, 2

[35] Feiping Nie, Rui Zhang, and Xuelong Li. A generalized power iteration method for solving quadratic problem on the stiefel manifold. *Sci. China Inf. Sci.*, 60(11):112101:1–112101:10, 2017. 5

[36] Yalan Qin, Chuan Qin, Xinpeng Zhang, and Guorui Feng. Dual consensus anchor learning for fast multi-view clustering. *IEEE Trans. Image Process.*, 33:5298–5311, 2024. 3

[37] Yalan Qin, Zhenjun Tang, Hanzhou Wu, and Guorui Feng. Flexible tensor learning for multi-view clustering with markov chain. *IEEE Trans. Knowl. Data Eng.*, 36(4):1552–1565, 2024. 3

[38] Elias Rajaby and Sayed Masoud Sayedi. A structured review of sparse fast fourier transform algorithms. *Digit. Signal Process.*, 123:103403, 2022. 4

[39] Ferdinand Samaria and Andy Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of Second IEEE Workshop on Applications of Computer Vision, WACV 1994, Sarasota, FL, USA, December 5-7, 1994*, pages 138–142. IEEE, 1994. 6, 2

[40] Allan C Spradling, Dianne Stern, Amy Beaton, E Jay Rhem, Todd Laverty, Nicole Mozden, Sima Misra, and Gerald M Rubin. The berkeley drosophila genome project gene disruption project: Single p-element insertions mutating 25% of vital drosophila genes. *Genetics*, 153(1):135–177, 1999. 6, 2

[41] Yinghui Sun, Xingfeng Li, Quansen Sun, Min-Ling Zhang, and Zhenwen Ren. Improved weighted tensor schatten p-norm for fast multi-view graph clustering. In *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 1427–1436. ACM, 2024. 3

[42] Martijn van Breukelen, Robert P. W. Duin, David M. J. Tax, and J. E. den Hartog. Handwritten digit recognition by combined classifiers. *Kybernetika*, 34(4):381–386, 1998. 6, 2

[43] Rong Wang, Jitao Lu, Yihang Lu, Feiping Nie, and Xuelong Li. Discrete and parameter-free multiple kernel k-means. *IEEE Trans. Image Process.*, 31:2796–2808, 2022. 1

[44] Rong Wang, Penglei Wang, Danyang Wu, Zhensheng Sun, Feiping Nie, and Xuelong Li. Multi-view and multi-order structured graph learning. *IEEE Trans. Neural Networks Learn. Syst.*, 35(10):14437–14448, 2024. 1

[45] Jie Wen, Zheng Zhang, Zhao Zhang, Lei Zhu, Lunke Fei, Bob Zhang, and Yong Xu. Unified tensor framework for incomplete multi-view clustering and missing-view inferring. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021*, pages 10273–10281. AAAI Press, 2021. 2, 4

[46] Jie Wen, Chengliang Liu, Gehui Xu, Zhihao Wu, Chao Huang, Lunke Fei, and Yong Xu. Highly confident local structure based consensus graph learning for incomplete multi-view clustering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 15712–15721. IEEE, 2023. 1

[47] Jie Wen, Zheng Zhang, Lunke Fei, Bob Zhang, Yong Xu, Zhao Zhang, and Jinxing Li. A survey on incomplete multiview clustering. *IEEE Trans. Syst. Man Cybern. Syst.*, 53(2): 1136–1149, 2023. 1

[48] Wei Xia, Quanxue Gao, Qianqian Wang, and Xinbo Gao. Tensor completion-based incomplete multiview clustering. *IEEE Trans. Cybern.*, 52(12):13635–13644, 2022. 1

[49] Wei Xia, Quanxue Gao, Qianqian Wang, Xinbo Gao, Chris Ding, and Dacheng Tao. Tensorized bipartite graph learning for multi-view clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):5187–5202, 2023. 1, 3, 4, 6

[50] Yuan Xie, Dacheng Tao, Wensheng Zhang, Yan Liu, Lei Zhang, and Yanyun Qu. On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *Int. J. Comput. Vis.*, 126(11):1157–1179, 2018. 1, 2, 5

[51] Changqing Zhang, Ehsan Adeli, Zhengwang Wu, Gang Li, Weili Lin, and Dinggang Shen. Infant brain development prediction with latent partial multi-view representation learning. *IEEE Trans. Medical Imaging*, 38(4):909–918, 2019. 1

[52] Changqing Zhang, Yajie Cui, Zongbo Han, Joey Tianyi Zhou, Huazhu Fu, and Qinghua Hu. Deep partial multiview learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44 (5):2402–2415, 2022. 1

[53] Zheng Zhang, Li Liu, Fumin Shen, Heng Tao Shen, and Ling Shao. Binary multi-view clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(7):1774–1782, 2019. 6, 3

[54] Zihua Zhao, Ting Wang, Haonan Xin, Rong Wang, and Feiping Nie. Multi-view clustering via high-order bipartite graph fusion. *Inf. Fusion*, 113:102630, 2025. 6, 3