Multi-view Ensemble Clustering via Low-rank and Sparse Decomposition: From Matrix to Tensor

XUANQI ZHANG, School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China
QIANGQIANG SHEN, School of Electronics and Information Engineering, Harbin Institute of Technology (Shenzhen), China
YONGYONG CHEN, School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China
GUOKAI ZHANG, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China
ZHONGYUN HUA and JINGYONG SU, School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China

As a significant extension of classical clustering methods, ensemble clustering first generates multiple basic clusterings and then fuses them into one consensus partition by solving a problem concerning graph partition with respect to the co-association matrix. Although the collaborative cluster structure among basic clusterings can be well discovered by ensemble clustering, most advanced ensemble clustering utilizes the self-representation strategy with the constraint of low-rank to explore a shared consensus representation matrix in multiple views. However, they still encounter two challenges: (1) ***high computational cost*** caused by both the matrix inversion operation and singular value decomposition of large-scale square matrices; (2) ***less considerable attention on high-order correlation*** attributed to the pursue of the two-dimensional pairwise relationship matrix. In this article, based on low-rank and sparse decomposition from both matrix and tensor perspectives, we propose two novel multi-view ensemble clustering methods, which tangibly decrease computational complexity. Specifically, our first method utilizes low-rank and sparse matrix decomposition to learn one common co-association matrix, while our last method constructs all co-association matrices into one third-order tensor to investigate the high-order correlation among multiple views by low-rank and sparse tensor decomposition. We adopt the alternating direction method of multipliers to solve two convex models

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62106063, by the Guangdong Natural Science Foundation under Grant No. 2022A1515010819, by the Shenzhen College Stability Support Plan under Grant No. GXWD20201230155427003-20200824113231001, by the Shenzhen Science and Technology Program under Grants No. RCBS20210609103708013 and No. JCYJ20220818102414031, by Humanities and Social Sciences Foundation of the Ministry of Education of China under Grant No. 22YJC630129, and by Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies under Grant No. 2022B1212010005.
Authors' addresses: X. Zhang, Y. Chen (corresponding author), Z. Hua, and J. Su, School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, 518055, China; emails: zxq5706346@gmail.com, YongyongChen.cn@gmail.com, huazyun@gmail.com, sujingyong@hit.edu.cn; Q. Shen, School of Electronics and Information Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen, 518055, China; email: 1120810623@hit.edu.cn; G. Zhang, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, 10001, China; email: zhangguokai_01@163.com.

1556-4681/2023/05-ART103 $15.00
https://doi.org/10.1145/3589768

by dividing them into several subproblems with closed-form solution. Experimental results on ten real-world datasets prove the effectiveness and efficiency of the proposed two multi-view ensemble clustering methods by comparing them with other advanced ensemble clustering methods.

CCS Concepts: • **Computing methodologies** → **Ensemble methods**; **Unsupervised learning**; *Spectral methods;*

Additional Key Words and Phrases: Multi-view clustering, ensemble clustering, low-rank and sparse decomposition

ACM Reference format:
Xuanqi Zhang, Qiangqiang Shen, Yongyong Chen, Guokai Zhang, Zhongyun Hua, and Jingyong Su. 2023. Multi-view Ensemble Clustering via Low-rank and Sparse Decomposition: From Matrix to Tensor. *ACM Trans. Knowl. Discov. Data.* 17, 7, Article 103 (May 2023), 19 pages.
https://doi.org/10.1145/3589768

1 INTRODUCTION

As a classic unsupervised learning algorithm, clustering analysis aims to segregate groups according to their similar characteristics and assign them into their corresponding clusters. Because of the difficulty and high cost of acquiring labeled data, clustering has been widely used in text categorization [11], image clustering [38], image compression [10], and video processing [45]. Till now, there are several kinds of clustering algorithms, e.g., K-means, kernel-based clustering, graph-based clustering, spectral clustering [25], and so on. Usually, each clustering algorithm has its strengths and drawbacks. For example, it is known that the K-means method is the basic fast one among all methods. Nevertheless, it is sensitive to the initialization of center points and noise. Although kernel-based methods handle non-linear data perfectly, their performance highly depends on the choice of kernel types such as Gaussian, linear, and polynomial. Thus, constructing a general clustering algorithm for different kinds of datasets from the real world is challenging. Confronted with this challenge, one naive way is to yield multiple base clusterings and fuse them into one consensus base for the final clustering result. Following this idea, ensemble clustering used different techniques such as the graph partitioning method to get a better and more consensus clustering and has been an efficient alternative to resolve cluster structures [17, 30].

Recently, researchers have proposed considerable ensemble clustering algorithms to capture various cluster structures of the data [17, 29, 30, 46], most of which adopted two steps, including **basic partitions (BPs)** generation and aggregation. That is, the first step aims to generate the co-association matrices derived from BPs, while the last one is to yield the consensus clustering result. Two kinds of strategies exist to fulfill the basic cluster integration [30]: designing some utility functions or constructing co-association matrices. The former tries to depict the similarity between different basic clusterings, while the latter contributes to transforming the similarity of basic clusterings into a graph partition problem. Taking co-association matrices methods as an example, Tao et al. [30] and Li et al. [17] have combined the ensemble clustering with the popular spectral clustering into one unified model. Their main differences are that the former work resorted to the self-representation property to uncover the low-rank representation with the block-diagonal structure, while the last work used the ensemble learning strategy to obtain a robust presentation of graph Laplacian matrix. Later, one nonconvex rank minimization model of Reference [30] was further developed in its journal paper [29]. Liu et al. [22] investigated the K-means method to reduce the high complexity of spectral ensemble clustering. However, there has been a growing concern about the performance of existing ensemble clustering methods when solving multi-view data, not single-view data. This is because multi-view data are ubiquitous in various applications, such as video surveillance, natural language processing, human action recognition, and so on.

To overcome this issue, many studies have been proposed by extending existing single-view ensemble clustering algorithms by investigating the consistency or/and diversity among basic partitions. One typical instance is Reference [27], which extended the single-view ensemble clustering method in Reference [30] by pursuing one low-rank common co-association matrix for consistency. To eliminate the data corruption and noises, the marginalized denoising process was integrated into the ensemble clustering in Reference [28]. Also, Liu et al. [20] formulated the consensus-guided multi-view clustering method into two k-means clusterings iteratively with high-efficiency [36].

In general, the majority of existing approaches may suffer from the following deficiencies: (1) ***High computational cost*:** Current advanced single-view and multi-view ensemble clustering algorithms blindly pursue performance improvement through self-representation strategy at high computational cost, such as in References [27–30]. The high computational cost attributes to the matrix inversion operation and singular value decomposition of $n \times n$ co-association matrices, where n denotes the number of samples. (2) ***High-order correlation loss*:** In the second step of aggregation, most existing multi-view ensemble clustering methods utilized graph, co-association, k-means to explore two-dimensional pair-wise relationship matrix from BPs, yet without exploring the high-order reciprocity in different views.

Regarding the above problems, rooted in the low-rank and sparse decomposition rather than the early self-representation framework, we propose two innovative multi-view ensemble clustering methods, abbreviated as **matrix multi-view ensemble clustering (MMEC)** and **tensor multi-view ensemble clustering (TMEC)** with both clustering performance improvement and computational cost guarantee. The flowchart of our proposed two multi-view ensemble clustering methods is shown in Figure 1. Different from current excellent multi-view ensemble clustering algorithms [27–30, 39–41, 43], which utilized the self-representation framework to explore the higher level information of BPs, the proposed MMEC and TMEC decompose co-association matrices or tensor into the low-rank term and sparse part (as shown in Figure 1(c)) without the matrix inversion operation, and thus yielding lower computation complexity. Moreover, in light of low-rank and sparse decomposition of matrices $S^{(i)}$, TMEC transforms all co-association matrices $\{S^{(i)}\}_{i=1}^{m}$ into tensor $\mathcal{S}$ to explore the high-level information and the high-order correlation among data with multi-view in the same time. To sum up, the contributions of the article are summarized as follows:

- We propose two novel multi-view ensemble clustering methods to explore heterogeneous and higher level information of multi-view data. The low-rank and sparse decomposition are utilized in learning co-association matrices in the multi-view ensemble clustering, which could significantly speed up the algorithm's operation.
- The first matrix-based method decomposes each co-association matrix, taking them as the collection of one shared low-rank matrix and one sparse noise matrix. The other tensor-based method stores all co-association matrices into the co-association tensor with the tensor constraint in low-rank to investigate the high-order correlation of multi-view data.
- We conducted experiments on ten real-world datasets to demonstrate the effectiveness of the proposed MMEC and TMEC methods compared to other highly sophisticated multi-view ensemble clustering models.

The rest of this article is organized as follows: We will briefly review ensemble clustering methods and the low-rank and sparse decomposition techniques and applications in Section 2. Some preliminary knowledge about ensemble clustering and tensor optimization is given in Section 3. Moreover, Section 4 gives two novel multi-view ensemble clustering methods based on the low-rank and sparse decomposition and their corresponding solutions. Last, the conclusion of this article is summarized in Section 5.

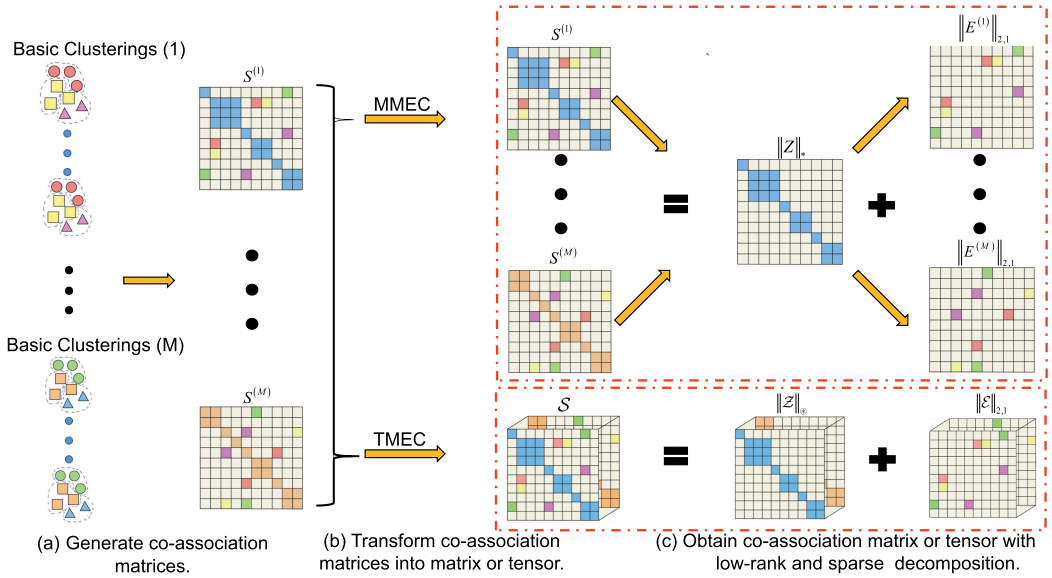

Fig. 1. The pipeline of our proposed two multi-view ensemble clustering algorithms, MMEC and TMEC. MMEC is based on the low-rank and sparse matrix decomposition while TMEC resorts to the low-rank and sparse tensor decomposition. Both of them follow three steps: (a) Let $\{X^{(i)}\}_{i=1}^{m}$ represent multi-view data with m views and n samples. For the ith view, we generate r basic partitions (BPs) following Random Parameter Selection (RPS) strategy and then fuse them as a co-association matrix $S^{(i)}$. (b) MMEC is without any processing while TMEC stores all co-association matrices $\{S^{(i)}\}_{i=1}^{m}$ into the co-association tensor $\mathcal{S}$. (c) MMEC decomposes each co-association matrix $S^{(i)}$ as the sum of low-rank matrix Z and the sparse noise matrix $E^{(i)}$ while TMEC decomposes $\mathcal{S}$ into the low-rank tensor $\mathcal{Z}$ and the sparse noise tensor $\mathcal{E}$. Finally, the low-rank co-association matrix Z and co-association tensor $\mathcal{Z}$ serve as the import of the spectral clustering method to acquire the final partition.

2 RELATED WORK

This section introduces related work, including ensemble clustering methods with respect to application and low-rank and sparse decomposition concerning technique.

2.1 Ensemble Clustering

Ensemble clustering refers to a combining process, which transforms multiple clustering models or partitions into a single strengthened division. The process is likewise defined as consensus clustering or aggregation clustering [1]. In principle, a compelling ensemble clustering method has the ability to create more accurate, steady, and conformable clustering results compared with the independent clustering methods.

From supervised learning to unsupervised learning, this transmission is not straightforward as the conception, because there are various unique and difficult problems when constructing ensemble clustering. Among them, the important and most challenging one is the way to merge multiple clusters developed by the single clustering methods. As this cannot be done through simple voting or average operation, more complicated aggregating approaches and schemes are required. Thus, researchers have proposed several strategies using voting, utility functions, and co-association matrices. For instance, Wang et al. [34] designed a soft-voting algorithm to combine multiple basic clusterings. Also, Topchy et al. [31] used the generalized mutual information definition and proposed a category utility function-based objective function for consensus clustering.

To combine multiple partitions, Strehl et al. [26] conducted three graph algorithms in ensemble clustering. Then, Fern et al. [12] improved the clustering performance with bipartite graph. These works first generated similarity matrices and then utilized ensemble algorithms to obtain the final results. In 2005, the co-association matrix was stated by Fred et al. First, hierarchical clustering methods are employed for the consensus partition [13], which specifies the ensemble clustering problem as a classic problem about graph partition. In terms of the co-association-based methods, Liu et al. [21] stated the spectral ensemble clustering method and transformed the classical graph partition problem into the weighted K-means clustering problem. To avoid input noises from basic clusterings and reveal the intrinsic cluster structure in basic clusterings, Tao et al. [29] designed a novel robust spectral ensemble clustering approach as the robust version of Reference [21] by learning a common representation Z in low-rank form for all co-association matrices. Moreover, Tao et al. first integrated the ensemble clustering problem with the spectral clustering theory to handle ensemble clustering tasks the from single-view and multi-view in Reference [27] and further added the marginalized denoising strategy to detect the noises in Reference [28].

However, most existing ensemble clustering approaches learned pair-wise relationship matrices from basic clusterings without analyzing the high-order correlation within diverse views. They mainly attempt to make denoising under the self-representation property. In contrast, our proposed models learn high-level information and high-order correlation among all views simultaneously to uncover the underlying correlation for better clustering, which highlights the block-diagonal structure of co-association matrices.

2.2 Low-rank and Sparse Decomposition

Recent years, the low-rank and sparse decomposition-based methods have been applied in various fields, such as color image processing [9], hyperspectral image restoration [7], hyperspectral anomaly detection [18], deep compression [42], multi-source heterogeneous domain adaptation [35], traffic event detection [15], and multi-view spectral clustering [8, 37]. For more applications, please refer to the excellent papers [2, 3, 32, 33] and the references therein.

Via the decomposition of low-rank and sparse matrix, Candes et al. proposed a novel **robust principal component analysis (RPCA)** model in 2009. Since the decomposition is non-parametric without making any postulate, it has been widely used in a large scale of problems [2]. In the image processing domain, it is efficient to distinguish information from noise, edge, and outliers with the method. Specifically, the decomposition with low-rank and the sparse property has well worked out image analysis like image denoising, image colorization, and face recognition. Moreover, decomposition of low-rank and sparse has been exploited in latent variable selection. For instance, Chandrasekaran et al. [6] utilized this decomposition to explore the latent components. The method aims to learn a statistical model that observes samples from a subset of random variables, which are based on the graphic attributes of low-rank and sparse matrices decomposition. Then, it gets the entire collection of variables. The recent advanced variants such as stable RPCA, inductive RPCA, enhanced RPCA have been reviewed in References [2, 3, 33].

In terms of the multi-view clustering problem, a novel Markov chain-based multi-view spectral clustering model was designed by Xia et al. [37] to find a real matrix of transition probability. In our proposed methods, we generate multiple co-association matrices or a third-order tensor by collecting multiple matrices in the decomposition of low-rank and sparse. Furthermore, we complete our formulation and optimization problems for learning co-association matrices.

3 PRELIMINARY

There are some frequently used notations and definitions given in this section. Also, we provide explanations that are helpful to understand our low-rank and sparse decomposition methods from

Table 1. Primary Notations and Corresponding Information

Notation	Meaning
$\mathcal{A}$, A, a	tensor, matrix, vector
$\mathcal{A}^{(k)}$	the kth frontal slice of tensor $\mathcal{A}$
$\mathcal{A}_{(i)}$	Mode-i matricization of $\mathcal{A}$
$\mathcal{A}_f = \mathbf{fft}(\mathcal{A}, [], 3)$	the fast Fourier transformation
n, M	the number of samples, views
$\mathcal{A}_{ijk}$	the (i, j, k)th entry of $\mathcal{A}$
$\mathcal{A}_{(i,:,:)}$	the ith horizontal slice of $\mathcal{A}$
$\mathcal{A}_{(:,i,:)}$	the ith lateral slice of $\mathcal{A}$
$\mathcal{A}_{(:,:,i)}$	the ith frontal slice of $\mathcal{A}$
$\mathcal{S} \in \mathbb{R}^{n\times n\times M}$	the co-association tensor
$E^{(v)} \in \mathbb{R}^{n\times n}$	the error matrix for vth view
$\|\cdot\|_{2,1}$, $\|\cdot\|_F$	$l_{2,1}$-norm, Frobenius norm
$\|\cdot\|_{\circledast}$, $\|\cdot\|_\infty$	t-SVD-nuclear norm, infinity norm

matrix to tensor. To indicate the tensors, matrices, and vectors, we use letters in calligraphy, capital, and lowercase (e.g., $\mathcal{A}$, A, a), respectively. The regularly employed notations are epitomized in Table 1.

Nowadays, many researchers have utilized the tensor nuclear norm [16] to represent the low-rank attribute of hyperspectral images, color images, grayscale videos, and so on. Thus, we first introduce the definition of the tensor nuclear norm. Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the definition of block circular matrix $\mathbf{bcirc}(\mathcal{A})$ and block diagonal matrix $\mathbf{bdiag}(\mathcal{A})$ are formulated as

$$\mathbf{bcirc}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(n_3)} & \cdots & \mathcal{A}^{(2)} \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & \cdots & \mathcal{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}^{(n_3)} & \mathcal{A}^{(n_3-1)} & \cdots & \mathcal{A}^{(1)} \end{bmatrix},$$

$$\mathbf{bdiag}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} & & & \\ & \mathcal{A}^{(2)} & & \\ & & \ddots & \\ & & & \mathcal{A}^{(n_3)} \end{bmatrix}.$$

Given $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2\times n_4\times n_3}$, the **t-product** outcome is an $n_1\times n_4\times n_3$ tensor defined as $\mathcal{A} * \mathcal{B}$. More specifically, $\mathcal{A} * \mathcal{B} = \mathbf{bvfold}(\mathbf{bcirc}(\mathcal{A}) * \mathbf{bvec}(\mathcal{B}))$. $\mathbf{bvec}(\mathcal{A}) = [\mathcal{A}^{(1)}; \cdots; \mathcal{A}^{(n_3)}]$ indicates the block vectorization, while $\mathbf{bvfold}(\mathbf{bvec}(\mathcal{A})) = \mathcal{A}$ and $\mathbf{bdfold}(\mathbf{bdiag}(\mathcal{A})) = \mathcal{A}$ indicates the inverse operations of **bvec** and **bdiag**, respectively. By transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 along n_3, we can get the **transpose** of $\mathcal{A}$, which is defined as $\mathcal{A}^T \in \mathbb{R}^{n_2\times n_1\times n_3}$. The first frontal slice of the **identity tensor** $\mathcal{I} \in \mathbb{R}^{n_1\times n_1\times n_3}$ is an $n_1 \times n_1$ identity matrix and the remaining frontal slices of it are zero. If a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_1\times n_3}$ has $\mathcal{A}^T * \mathcal{A} = \mathcal{A} * \mathcal{A}^T = \mathcal{I}$, then the tensor is **orthogonal**.

Definition 1 (Co-association Matrix). For $\forall v$, given a set of data points $X^{(v)} = \{x_1^{(v)}, \ldots, x_n^{(v)}\} \in \mathbb{R}^{d_v\times n}$, which are sampled from K clusters $C = \{C_1, \ldots, C_K\}$, $1 \le v \le M$. Let $\Pi^{(v)} = \{\pi_1^{(v)}, \ldots, \pi_r^{(v)}\}$ represent a group of r BPs for $X^{(v)}$, $1 \le \pi_i^{(v)}(x_j^{(v)}) \le K_i$, $1 \le i \le r$, $1 \le j \le n$, $\pi_i^{(v)} = \{\pi_i^{(v)}(x_1^{(v)}), \ldots, \pi_i^{(v)}(x_n^{(v)})\}$ is one of the categories, which partitions $X^{(v)}$ into K_i clusters.

To summarize $\Pi^{(v)}$, we can define the co-association matrix $S^{(v)} \in \mathbb{R}^{n\times n}$:

$$S^{(v)}\left(x_a^{(v)}, x_b^{(v)}\right) = \frac{1}{r}\sum_{i=1}^{r}\delta\left(\pi_i^{(v)}\left(x_a^{(v)}\right), \pi_i^{(v)}\left(x_b^{(v)}\right)\right), \tag{1}$$

where $x_a^{(v)}, x_b^{(v)} \in X^{(v)}$. $\delta(a, b)$ is 1 if $a = b$; 0 otherwise.

*Definition 2 (**Tensor-Singular Value Decomposition (t-SVD)** [16]).* Given $\mathcal{X}$, the definition of its t-SVD is

$$\mathcal{X} = \mathcal{U} * \mathcal{G} * \mathcal{V}^T.$$

The orthogonal tensors are defined as $\mathcal{U} \in \mathbb{R}^{n_1\times n_1\times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2\times n_2\times n_3}$, while $\mathcal{G} \in \mathbb{R}^{n_1\times n_2\times n_3}$ serves as an f-diagonal tensor. For all the frontal slices, they are diagonal matrices.

According to the t-SVD, the tensor nuclear norm can be described as follows:

Definition 3 (Tensor Nuclear Norm [16]). Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, as the sum of singular values of all the frontal slices of $\hat{\mathcal{X}}$, the tensor nuclear norm of the tensor is defined as $\|\mathcal{X}\|_{\circledast}$,

$$\|\mathcal{X}\|_{\circledast} = \sum_{i=1}^{\min\{n_1, n_2\}}\sum_{j=1}^{n_3}|\hat{\mathcal{G}}(i, i, j)|, \tag{2}$$

where $\hat{\mathcal{X}}$ indicates the Discrete Fourier Transformation of $\mathcal{X}$. Within the unit ball of the tensor spectral norm, the above tensor nuclear norm is the convex envelope of the tensor average rank.

4 PROBLEM FORMULATION

The proposed two multi-view ensemble clustering methods are abbreviated as MMEC and TMEC, respectively. As shown in Figure 1, MMEC uses the matrix decomposition in low-rank and sparse matrix while TMEC adopts the low-rank and sparse tensor decomposition. First, we introduce the formulation of the MMEC method. Second, we will illustrate our TMEC model, which is an improvement of the matrix-based method MMEC.

4.1 The First Matrix-based Model: Formulation of MMEC

Given multi-view data $\{X^{(v)}\}_{v=1}^{M}$ and following the previous methods [27, 29], we generate multiple basic partitions by the random parameter selection strategy for each view individually and then compute the co-association matrices $S^{(1)}, S^{(2)}, \ldots, S^{(M)}$ by Equation (1) to investigate the higher level information from BPs. To find a consensus clustering result, Tao et al. [27] proposed the following model to learn a general representation Z employed by multi-view co-association matrices:

$$\begin{aligned}&\min_{H, Z, E^{(v)}} tr(H^T L_Z H) + \lambda_1\|Z\|_* + \lambda_2\sum_{v=1}^{M}\|E^{(v)}\|_1\\ &s.t.\ S^{(v)} + HH^T = S^{(v)}Z + E^{(v)}, Z \geq 0,\ Z\mathbf{1} = \mathbf{1},\ H^T H = I,\end{aligned} \tag{3}$$

where $S^{(v)} \in \mathbb{R}^{n\times n}$ is constructed by Equation (1). $Z \in \mathbb{R}^{n\times n}$ denotes the shared low-rank representation among all views. $E^{(v)} \in \mathbb{R}^{n\times n}$ represents the noises of the v-view. The first term concerning the objective function in Equation (3) stems from the spectral clustering, which is based on the Laplacian matrix L_Z. $H \in \mathbb{R}^{n\times K}$ denotes the unified indicator matrix for consensus clustering result. $\|Z\|_*$ denotes the matrix nuclear norm, which is denoted as the sum of non-zero singular values of Z. In addition, $\|E\|_1 = \sum_{i,j} e_{i,j}$ is the l_1 norm.

Although the complementary information can be well explored in the above method and its enhanced version [28], there may exist two shortcomings: (1) These methods must perform the

matrix inversion operator on the $n \times n$ matrix, the singular value decomposition from the low-rank matrix approximation, and the eigenvalue decomposition on the variable H, all of which take expensive time. (2) They focus on finding the common two-dimensional representation Z of different views, but unfortunately fail to discover the high-order interaction between multiple views. To conquer the first challenge, we decompose each co-association matrix Z into two parts: one shared latent co-association matrix Z that indicates the implicit real clustering structure and an error matrix $E^{(v)}$ that encodes the noise in each view:

$$S^{(v)} = Z + E^{(v)}, \ v = 1, \ldots, M. \tag{4}$$

Inspired by Equation (3), we also exploit the matrix nuclear norm as the regularizer to model the latent matrix Z. But, we adopt the $l_{2,1}$ norm instead of the l_1 norm, since the difference between $S^{(v)}$ and Z could be tiny [19]. Consequently, under the low-rank and sparse assumption, our proposed matrix-based multi-view ensemble clustering method MMEC is formulated as

$$\begin{aligned} &\min_{Z, E^{(v)}} \|Z\|_* + \lambda \sum_{i=1}^{M} \|E^{(v)}\|_{2,1} \\ &\text{s.t. } S^{(v)} = Z + E^{(v)}, \ v = 1, \ldots, M, \end{aligned} \tag{5}$$

where $\|E^{(v)}\|_{2,1}$ is defined as $\sum_j \|E^{(v)}(:, j)\|_2$. λ is a weight parameter balancing the costs of low-rank terms and sparse terms. In our MMEC model Equation (5), it follows the low-rank and sparse decomposition strictly rather than the self-representation framework used in most advanced multi-view ensemble clustering methods such as References [27–30], and thus avoiding matrix inversion operation perfectly. The low-rank term $\|Z\|_*$ could reveal the pair-wise membership under a multi-view setting and bring out the same cluster information shared by multiple views [28]. The $l_{2,1}$ norm has been proven to remove noise such as hyperspectral image denoising [7, 14].

4.2 The Second Tensor-based Model: Formulation of TMEC

Our first MMEC model has addressed the first shortcoming of ***high computational cost*** by strictly following the decomposition of low-rank and sparse matrix. The shared low-rank representation Z explores the consistency of multiple views features only, without considering the internal diversity in each view [24]. For the second limitation, we investigate the tensor optimization to uncover the high-order correlation in all views via storing every $S^{(v)}$ as the vth frontal slice of third-order tensor $\mathcal{S} \in \mathbb{R}^{n \times n \times M}$ as shown in Figure 1(c). Then, we separate $\mathcal{S}$ into one tensor $\mathcal{Z}$ and one noise tensor $\mathcal{E}$, i.e.,

$$\mathcal{S} = \mathcal{Z} + \mathcal{E}. \tag{6}$$

Compared with Equation (4), which uses the matrix optimization, Equation (6) utilizes the tensor optimization, which explores the view dimension explicitly, thereby possibly achieving the high-order correlation.

Different $Z^{(v)}$ contain some related information as the multi-view characters are obtained from identical objects. In addition, it is noted that the amount of samples is much bigger than the amount of clusters. As a consequence, the tensor $\mathcal{Z}$ is supposed to be low-rank. To obtain and regularize the original objective function for our model, We exploit the tensor nuclear norm based on t-SVD in Equation (2). Therefore, our second tensor-based method is formulated as

$$\begin{aligned} &\min_{\mathcal{Z}, \mathcal{E}} \|\mathcal{Z}\|_{\circledast} + \lambda \|\mathcal{E}\|_{2,1} \\ &\text{s.t. } \mathcal{S} = \mathcal{Z} + \mathcal{E}, \end{aligned} \tag{7}$$

where $\mathcal{S}$ denotes co-association tensor. In Section 5, we will observe that (1) our second tensor-based method TMEC could significantly improve the ensemble clustering performance over our

ALGORITHM 1: MMEC for multi-view ensemble clustering

Input: Co-association matrices $\{S^{(1)}, \ldots, S^{(M)}\}$, trade-off parameter λ;
Initialize: $Z_0, E_0^{(v)}, Y_0^{(v)}$ initialized to $\mathbf{0}$; $k = 0$, $\mu_0 = 10^{(-6)}$, $\rho = 2$, $\mu_{max} = 10^{10}$, $\epsilon = 10^{-8}$;

```
while not converged do
    Update the shared low-rank representation Z_{k+1} by Equation (10);
    for v = 1 to M do
        Update the noise matrix E_{k+1}^{(v)} by Equation (13);
        Update Lagrangian multiplier Y_{k+1}^{(v)} by Equation (15);
    end for
    Update the penalty parameter μ_{k+1} = min(ρμ_k, μ_max);
    Check the condition of convergence:
```

$$\max \left\{ \begin{array}{l} \|Z_{k+1} - Z_k\|_\infty \\ \|E_{k+1}^{(v)} - E_k^{(v)}\|_\infty,\ v = 1, 2, \ldots, M \\ \|Z_{k+1} + E_{k+1}^{(v)} - S^{(v)}\|_\infty,\ v = 1, 2, \ldots, M \end{array} \right\} \leq \epsilon; \tag{9}$$

```
end while
```

Output: Z_k, $E_k^{(v)} (v = 1, 2, \ldots, M)$.

first matrix-based method MMEC, and (2) TMEC not only avoids the high computation cost but also investigates the high-order correlation across multiple views.

4.3 Optimization of MMEC

Based on the problem formulation, the optimization problem could be settled, since the objective function includes the convex matrix nuclear norm and the convex $l_{2,1}$ norm. Thus, we solve the MMEC model in Equation (5) through the **Alternating Direction Method of Multipliers (ADMM)** scheme. The corresponding augmented Lagrange function of Equation (5) is

$$\mathcal{L}(Z, E^{(v)}) = \|Z\|_* + \lambda \sum_{v=1}^{M} \|E^{(v)}\|_{2,1} + \sum_{v=1}^{M} \langle Y^{(v)}, Z + E^{(v)} - S^{(v)} \rangle + \frac{\mu}{2} \sum_{v=1}^{M} \|Z + E^{(v)} - S^{(v)}\|_F^2, \tag{8}$$

where $Y^{(i)} \in \mathbb{R}^{n \times n}$ denotes the Lagrangian multiplier corresponding to the vth view. $\langle A, B \rangle$ indicates the inner product of matrices defined as $tr(A^T B)$, and the variable μ is an adaptive penalty parameter.

The description of our designed algorithm based on the matrix in low-rank and sparse decomposition is presented in Algorithm 1. Then, we would explain the update of Z and $E^{(v)}$.

1. Z-subproblem:

While other variables are settled, the subproblem about Z is

$$\begin{aligned} Z_{k+1} &= \arg\min_Z \|Z\|_* + \sum_{v=1}^{M} \langle Y_k^{(v)}, Z + E_k^{(v)} - S^{(v)} \rangle + \frac{\mu_k}{2} \sum_{v=1}^{M} \|Z + E_k^{(v)} - S^{(v)}\|_F^2 \\ &= \arg\min_Z \|Z\|_* + \frac{\mu_k}{2} \sum_{v=1}^{M} \left\| Z + E_k^{(v)} - S^{(v)} + \frac{Y_k^{(v)}}{\mu_k} \right\|_F^2 \\ &= \arg\min_Z \|Z\|_* + \frac{\mu_k M}{2} \left\| Z - \frac{1}{M} \sum_{v=1}^{M} \left(S^{(v)} - E_k^{(v)} - \frac{Y_k^{(v)}}{\mu_k} \right) \right\|_F^2, \end{aligned} \tag{10}$$

which can be solved by the method of singular value threshold [5]. Suppose that the SVD form of $(\frac{1}{M}\sum_{v=1}^{M}(S^{(v)} - E_k^{(v)} - \frac{Y_k^{(v)}}{\mu_k}))$ is $U\Sigma V^T$, the solution to Equation (10) is as follows:

$$Z_{k+1} = U\mathcal{T}_{1/\mu M}(\Sigma)V^T, \tag{11}$$

where the shrinkage operator is $\mathcal{T}_\lambda(A) = \max(A - \lambda, 0) + \min(A + \lambda, 0)$.

2. $E^{(v)}$-subproblem:

We fix other variables and then the subproblem with respect to $E^{(v)}$ can be reduced as

$$\begin{aligned} E_{k+1}^{(v)} &= \arg\min_{E^{(v)}} \lambda \sum_{v=1}^{M} \|E^{(v)}\|_{2,1} + \sum_{v=1}^{M} \langle Y_k^{(v)}, Z_{k+1} + E^{(v)} - S^{(v)} \rangle + \frac{\mu_k}{2} \sum_{v=1}^{M} \|Z_{k+1} + E^{(v)} - S^{(v)}\|_F^2 \\ &= \arg\min_{E^{(v)}} \sum_{v=1}^{M} \left(\lambda \left\|E^{(v)}\right\|_{2,1} + \frac{\mu_k}{2} \left\| E^{(v)} - \left(S^{(v)} - Z_{k+1} - \frac{Y_k^{(v)}}{\mu_k} \right) \right\|_F^2 \right). \end{aligned} \tag{12}$$

We can see that the $E^{(v)}$-subproblem is independent of other $(M-1)$ variables, which means that the $E^{(v)}$-subproblem can be solved by

$$E_{k+1}^{(v)} = \arg\min_{E^{(v)}} \frac{\lambda}{\mu_k}\|E^{(v)}\|_{2,1} + \frac{1}{2}\left\| E^{(v)} - \left(S^{(v)} - Z_{k+1} - \frac{Y_k^{(v)}}{\mu_k} \right) \right\|_F^2. \tag{13}$$

As proven in Reference [19], the above subproblem has the closed-form solution, the jth column of which is

$$E_{k+1}^{(v)}(:, j) = \begin{cases} \frac{\|H_k^{(v)}(:,j)\|_2 - \frac{\lambda}{\mu_k}}{\|H_k^{(v)}(:,j)\|_2} H_k^{(v)}(:, j), & \text{if } \frac{\lambda}{\mu_k} < \|H_k^{(v)}(:, j)\|_2, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where $H_k^{(v)} = S^{(v)} - Z_{k+1} - \frac{Y_k^{(v)}}{\mu_k}$.

3. Update multipliers: We update each Lagrangian multiplier by

$$Y_{k+1}^{(v)} = Y_k^{(v)} + \mu_k \left(Z_{k+1} + E_{k+1}^{(v)} - S^{(v)} \right). \tag{15}$$

Once the shared low-rank representation Z is learned from Algorithm 1, we can use Z for the input co-association matrix to the MMEC algorithm for the spectral clustering method and eventually get the clustering solution.

4.4 Optimization of TMEC

Similar to Equation (5), our TMEC model in Equation (7) is also a convex optimization problem and thus, we adopt ADMM to solve it. Mathematically, the augmented Lagrangian function of Equation (7) is defined as

$$\begin{aligned} \mathcal{L}(\mathcal{Z}, \mathcal{E}) &= \|\mathcal{Z}\|_{\circledast} + \lambda\|\mathcal{E}\|_{2,1} + < \mathcal{Y}, \mathcal{S} - \mathcal{Z} - \mathcal{E} > + \frac{\mu^k}{2}\|\mathcal{S} - \mathcal{Z} - \mathcal{E}\|_F^2 \\ &= \|\mathcal{Z}\|_{\circledast} + \lambda\|\mathcal{E}\|_{2,1} + \frac{\mu}{2}\|\mathcal{S} - \mathcal{Z} - \mathcal{E} + \mathcal{Y}/\mu\|_F^2, \end{aligned} \tag{16}$$

where $\mathcal{Y} \in \mathbb{R}^{n \times n \times M}$ represents the Lagrangian multiplier.

1. $\mathcal{Z}$-subproblem: In the iterative process, the optimal low-rank tensor representation $\mathcal{Z}$ can be updated by the optimization as follows:

$$\mathcal{Z}_{k+1} = \arg\min_{\mathcal{Z}} \frac{1}{\mu_k}\|\mathcal{Z}\|_{\circledast} + \frac{1}{2}\|\mathcal{Z} - (\mathcal{S} - \mathcal{E}_k + \mathcal{Y}_k/\mu_k)\|_F^2. \tag{18}$$

ALGORITHM 2: TMEC for multi-view ensemble clustering

Input: Co-association matrices $\{S^{(1)}, \ldots, S^{(M)}\}$, trade-off parameter λ;
Initialize: $\mathcal{Z}_0, \mathcal{E}_0, \mathcal{Y}_0$ initialized to $\mathbf{0}$; $k = 0$, $\mu_0 = 10^{(-6)}$, $\rho = 2$, $\mu_{max} = 10^{10}$, $\epsilon = 10^{-6}$;

1: **while** not converged **do**
2: Update the low-rank tensor $\mathcal{Z}_{k+1}$ by Equation (18);
3: Update the sparse noise tensor $\mathcal{E}_{k+1}$ by Equation (20);
4: Update Lagrangian multiplier $\mathcal{Y}_{k+1}$ by Equation (23);
5: Update the penalty parameter $\mu_{k+1} = \min(\rho\mu_k, \mu_{max})$;
6: Check the conditions of convergence:

$$\max \left\{ \begin{array}{l} \|\mathcal{Z}_{k+1} - \mathcal{Z}_k\|_\infty \\ \|\mathcal{E}_{k+1} - \mathcal{E}_k\|_\infty \\ \|\mathcal{S} - \mathcal{Z}_{k+1} - \mathcal{E}_{k+1}\|_\infty \end{array} \right\} \leq \epsilon; \tag{17}$$

7: **end while**
Output: $\mathcal{Z}_k, \mathcal{E}_k$.

Based on t-SVD, this is a tensor nuclear norm minimization problem with a closed-form solution, which employs the tensor tubal-shrinkage operator [16]:

$$\mathcal{Z}_{k+1} = C_\tau(\mathcal{H}_k) = \mathcal{U} * C_\tau(\mathcal{G}) * \mathcal{V}^T, \tag{19}$$

where $\mathcal{H}_k = \mathcal{S} - \mathcal{E}_k + \mathcal{Y}_k/\mu_k$ and its t-SVD is $\mathcal{H}_k = \mathcal{U}\mathcal{G}\mathcal{V}^T$. $\tau = \frac{1}{\mu_k}$. $C_\tau(\mathcal{G}) = \mathcal{G} * \mathcal{J}$, where $\mathcal{J}$ is an $n \times n \times M$ f-diagonal tensor, and the diagonal element in the Fourier domain of it is $\bar{\mathcal{J}}(i,i,j) = \max(1 - \frac{\tau}{\bar{G}(i,i,j)}, 0)$.

2. $\mathcal{E}$-subproblem: The sparse noise tensor $\mathcal{E}$ is updated by the following optimization issue:

$$\mathcal{E}_{k+1} = \arg\min_{\mathcal{E}} \frac{\lambda}{\mu_k}\|\mathcal{E}\|_{2,1} + \frac{1}{2}\|\mathcal{E} - (\mathcal{S} - \mathcal{Z}_{k+1} + \mathcal{Y}_k/\mu_k)\|_F^2. \tag{20}$$

As the $l_{2,1}$-norm of the tensor $\mathcal{E}$ is represented as the sum of l_2-norm for each mode-3 fiber, we unfold every tensor along the third mode, leading to the optimization problem as follows:

$$E_{(3)k+1} = \arg\min_{E_{(3)}} \lambda\|E_{(3)}\|_{2,1} + \frac{\mu_k}{2}\|E_{(3)} - (S_{(3)} - Z_{(3)(k+1)} + Y_{(3)k}/\mu_k)\|_F^2. \tag{21}$$

Let $C = S_{(3)} - Z_{(3)(k+1)} + Y_{(3)k}/\mu_k$, it has the following closed-form solution:

$$E_{(3)(:,j)} = \begin{cases} \frac{\|C_{:,j}\|_2 - \frac{\lambda}{\mu^k}}{\|C_{:,j}\|_2} C_{:,j}, & \text{if } \frac{\lambda}{\mu_k} < \|C_{:,j}\|_2; \\ 0, & \text{otherwise,} \end{cases} \tag{22}$$

where $C(:,j)$ indicates the jth column of the matrix C. We will transform $E_{(3)k+1}$ into tensor $\mathcal{E}_{k+1}$.

3. Update multipliers:

$$\mathcal{Y}_{k+1} = \mathcal{Y}_k + \mu^k(\mathcal{S} - \mathcal{Z}_{k+1} - \mathcal{E}_{k+1}). \tag{23}$$

We learn the tensor $\mathcal{Z}$ with low-rank property and the sparse tensor $\mathcal{E}$ iteratively as summarized in Algorithm 2. Finally, we obtain results of the clustering algorithm by running the spectral clustering method on $A = \frac{1}{M}\sum_v(|Z^{(v)}| + |Z^{(v)^T}|)$.

Table 2. Descriptions of the Real Multi-view Datasets

Database	Instance/Number	View1	View2	View3	View4	View5	View6
100leaves	1,600/100	64d	64d	64d	—	—	—
BBC4view	685/5	4,659d	4,633d	4,655d	4,684d	—	—
Flowers	1,360/17	1,360d	1,360d	1,360d	—	—	—
MSRC-v1	210/7	24d	576d	512d	256d	254d	—
Mfeat	2,000/10	216d	76d	64d	6d	240d	47d
Notting-Hill	4,660/5	6,750d	2,000d	3,304d	—	—	—
ORL	400/40	4,096d	3,304d	6,750d	—	—	—
3Sources	169/3	3,560d	3,631d	3,068d	—	—	—
20Newsgroups	500/3	2,000d	2,000d	2,000d	—	—	—
Caltech101-7	1,474/7	48d	40d	254d	1,984d	512d	928d

4.5 Convergence and Complexity

Convergence: Since the matrix nuclear norm, the tensor nuclear norm, and the $l_{2,1}$ norm are convex, they satisfy the general assumption of ADMM strictly. References [4, 23] have reported the proof of the global convergence of ADMM for the decomposition of low-rank and sparse with respect to matrix and tensor. In the next section, we will show the numerical convergence of MMEC and TMEC methods.

Complexity: As for the MMEC method, as shown in Algorithm 1, supposing the data is $n \times n$ with M views, the computing cost about the stated method contains these parts: (a) For the spectral clustering operation, it costs $O(n^3)$; (b) it will take $O(Mn^2)$ to update $E^{(i)}$ for each view; (c) updating Z costs $O(Mn^2)$ due to the SVD operation. The final product is accessed by performing the spectral clustering algorithm with a complexity of $O(n^3)$. Overall, the cost of Algorithm 1 is $O(2KMn^2 + 2n^3)$ in total, where K denotes the number of iterations.

For the tensor-based method TMEC, suppose the number of iterations is K and the tensor is $n \times n \times M$. The computing complexity of TMEC includes the following parts: (a) With a thresholding solution, $\mathcal{E}$ can be updated with complexity of $O(Mn^2)$. (b) As for updating $\mathcal{Z}$, we first rotate $\mathcal{Z}$ from $n \times n \times M$ to $n \times M \times n$ and then calculate the FFT and inverse FFT, which take $O(Mn^2 log(n))$. Since we know $n >> M$, the rotation can reduce the complexity from $O(Mn^3)$ to $O(Mn^2 log(n))$. The tensor tubal-shrinkage operator performs the singular value decomposition with the complexity of $O(M^2n^2)$. The complexity of learning the low-rank tensor $\mathcal{Z}$ and the sparse tensor $\mathcal{E}$ in Algorithm 2 is $O(KMn^2(M + log(n) + 1))$. The final product is accessed by performing the spectral clustering algorithm with a complexity of $O(n^3)$. In conclusion, the general complexity in Algorithm 2 is $O(KMn^2(M + log(n)) + n^3)$.

5 EXPERIMENT

5.1 Experimental Settings

We report and analyze the experiment results of our two methods, MMEC in Algorithm 1 and TMEC in Algorithm 2. Also, we compare our models with other recently proposed and advanced clustering algorithms on ten real world datasets by the evaluation of two popular validation criteria. Moreover, extensive discussion including parameter analysis, numerical convergence, and running time, is given at the end of this section.

(1) *Datasets*: We introduce ten datasets with multiple views, which are summarized in Table 2, including image data and text data.

The introductions of these datasets are indicated as follows:

- **100leaves**[1] is a dataset in the UCI repository. Each sample is one of the 100 leaves species. It consists of 1,600 instances with 3 views.
- **BBC4view**[2] is the News article dataset. There are 685 samples in total with 5 categories. BBC4view extracts 4 views.
- **Flowers**[3] includes 1,360 flower instances with 17 directories. There are three views known as 1,360d color, 1,360d texture, and 1,360d shape.
- **MSRC-v1**[4] contains 210 images extracted from 7 distinct subjects. As for the 5 multiple views, they are 5 different types of features like the LBP feature and the HOG feature.
- **Mfeat**[5] is a dataset with instances of handwritten numbers sampled from Dutch utility maps. The dataset has 2,000 instances and 10 cluster numbers.
- **Notting-Hill** [27] is a movie data set that contains 4,660 faces from 5 persons, represented as 4,660 instances and 5 categories.
- **ORL**[6] holds 400 images extracted from 40 diverse subjects. This dataset holds 40 directories, which stands for 40 different people. There are 10 images in each directory.
- **3Sources** [28] is a text dataset from three online news sources. This dataset contains 169 instances and 3 categories.
- **20Newsgroups**[7] comprises around 18,000 newsgroups posts on 20 topics. This dataset has 500 instances.
- **Caltech101-7**[8] consists of 7 categories with 1,474 images, and the dataset contains 6 different views.

(2) *Compared Methods*: As for the compared methods, there are four single-view clustering methods (**SC**, **rBDLR**, **SEC**, **RSEC**) and three multi-view clustering methods (**RMSC**, **MVEC**, **MV^{2}EC**). Specifically,

- **SC:** We conducted spectral clustering [25] in two models: **SC**$_{bsv}$ returns performance of the best single view while **SC**$_{sum}$ collected all views of the dataset and employed spectral clustering.
- **rBDLR**$_{bsv}$ returns the best result from a single view via **rBDLR** [44].
- **RMSC:** Robust Multi-view Spectral Clustering [37] is a highly sophisticated method based on Markov chain.
- **SEC:** Spectral Ensemble Clustering [21] is one of the popular ensemble clustering algorithms.
- **RSEC:** Robust Spectral Ensemble Clustering [29] is a single-view ensemble clustering model by investigating one shared representation in low-rank.
- **MVEC:** Multi-view Ensemble Clustering [27] extends **RSEC** for handling clustering in multiple views.
- **MV^{2}EC:** Marginalized ensemble clustering for multiple views is an ensemble clustering method derived from **MVEC**, which alleviates the noises of BPs by the marginalized denoising process [28].

[1] https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set.
[2] http://mlg.ucd.ie/datasets/segment.html.
[3] http://www.robots.ox.ac.uk/vgg/data/flowers/.
[4] http://research.microsoft.com/en-us/projects/objectclassrecognition/.
[5] https://archive.ics.uci.edu/ml/datasets/Multiple+Features.
[6] https://cam-orl.co.uk/facedatabase.html.
[7] http://qwone.com/~jason/20Newsgroups/.
[8] http://www.vision.caltech.edu/datasets/.

Note that **SEC** and **RSEC** are recently proposed single-view ensemble clustering methods while **MVEC** and **MV^{2}EC** are recently proposed multi-view ensemble clustering methods.

For a proper comparison, the **random parameter selection (RPS)** strategy [13] is selected to yield a set of 100 basic partitions for each view individually for all ensemble clustering methods including our MMEC and TMEC. Also, the parameters in compared methods are set as same as these papers suggested. The parameters we used are as follows: **SC:** We run 5 times with different settings and investigate the average performance with standard deviation. **SEC:** 100 sets of basic clusterings as the basic partitions in ensemble clustering. **RMSC:** the parameter λ of this method ranges $[0.005, 0.01, 0.05, 0.1, 0.5, 1]$. **RSEC:** parameters are selected as $\lambda_1 = 0.001$, $\lambda_2 = 0.01$. **MVEC:** parameters are selected as $\lambda_1 = 1$, $\lambda_2 = 0.01$. **MV^{2}EC:** there are 3 parameters settled as $\lambda_1 = 1, \lambda_2 = [0.01 : 0.1], \lambda_3 = 1$. **MMEC:** one parameter range is settled as $\lambda = [0.01 : 0.01 : 0.5]$. **TMEC:** the method tunes one parameter λ from $[0.01 : 0.01 : 0.1]$.

(3) *Evaluation Metrics:* By following [27], for the evaluation metrics, we use **accuracy (ACC)** and **normalized mutual information (NMI)** to evaluate the effectiveness of the designed methods. The definitions of ACC and NMI are as follows:

ACC is defined as a fraction about labels of cluster and the labels of ground-truth. With a dataset $\mathcal{X}$, which has n samples with C clusters, ACC can be defined as

$$ACC = \max_{f} \frac{1}{n} \sum_{i=1}^{n} \delta(y_i, f(\pi(x_i))), \tag{24}$$

where $1 \leq i \leq n$, $y_i \in [1, C]$ indicates the label of the ground truth for x_i, $x_i \in \mathcal{X}$, $\pi(x_i) \in [1, C]$, π represented as the result after clustering that match $\mathcal{X}$ with a label set $\{\pi(x_1), \ldots, \pi(x_n)\}$. To maximize the result of *ACC*, we utilize $f(\cdot)$ as permutation and search $f(\cdot)$ from each permutation of C labels of clusters.

NMI investigates the shared entropy of information among cluster labels and the ground truth, which is formulated as

$$NMI = \frac{\sum_a \sum_b n_{a,b} \log(\frac{n n_{a,b}}{n_a n_b})}{\sqrt{(\sum_a n_a \log \frac{n_a}{n})(\sum_b n_b \log \frac{n_b}{n})}}, \tag{25}$$

where C_a is obtained by the result of a partition, C_b is extracted from the ground truth, and n_a and n_b denote the amount of instances with the corresponding cluster. Moreover, $n_{a,b}$ is the amount of instances in both C_a and C_b.

Relative Error is used to act as the stop criteria for convergence, which is formulated as

$$Relative\ Error = \max \left\{ \begin{array}{l} \|\mathcal{Z}_{k+1} - \mathcal{Z}_k\|_\infty, \\ \|\mathcal{E}_{k+1} - \mathcal{E}_k\|_\infty, \\ \|\mathcal{S} - \mathcal{Z}_{k+1} - \mathcal{E}_{k+1}\|_\infty \end{array} \right\} \leq tol, \tag{26}$$

where $tol > 0$ is a parameter that is pre-determined.

5.2 Experimental Results

In this section, we describe the results of clustering from the accuracy perspective and time perspective.

(1) *Clustering Performance:* Generally, the results of all competing clustering models and our proposed MMEC and TMEC are summarized in Tables 3 and 4, in which the results with best performance are written in red and the ones with second-best performance are written in blue. According to Tables 3 and 4, we can get the following observations:

Table 3. Performance of Clustering (ACC%±std%) on Ten Real-world Datasets

Datasets	100leaves	BBC4view	Flowers	MSRC-v1	Mfeat	Notting-Hill	ORL	3Sources	20Newsgroups	Caltech101-7
SC_{bsv} [25]	47.38±1.04	34.41±0.52	28.82±0.75	54.76±1.48	66.60±0.09	71.78±0.00	57.96±0.25	63.15±0.78	47.54±1.22	44.50±0.30
SC_{sum} [25]	34.88±1.12	35.18±1.23	17.13±0.76	37.62±1.44	48.70±0.25	87.17±0.01	46.27±0.50	61.24±0.64	47.65±0.23	41.31±0.28
$rBDLR_{bsv}$ [44]	37.56±1.13	37.37±0.00	32.51±1.03	51.05±0.78	61.83±0.03	31.19±0.00	63.80±0.04	61.50±0.05	31.16±0.34	57.68±0.12
RMSC [37]	77.34±4.95	77.52±0.32	33.72±1.89	75.24±4.78	76.08±5.67	78.04±4.92	70.00±0.03	58.30±0.02	89.82±0.08	39.17±0.02
SEC_{bsv} [21]	53.84±1.90	81.75±0.00	34.09±0.61	68.95±0.62	86.20±0.40	83.69±0.00	55.70±2.25	53.58±1.94	23.80±1.22	53.85±1.13
SEC_{sum} [21]	60.58±0.25	83.80±0.00	39.78±0.23	67.71±1.03	84.00±0.00	95.21±0.00	47.25±0.82	56.80±3.50	35.40±2.22	54.99±1.45
$RSEC_{bsv}$ [29]	23.44±0.00	31.97±6.57	22.57±0.00	67.72±0.00	59.75±0.00	74.91±0.00	63.00±1.33	65.72±0.00	21.20±0.35	54.27±0.44
$RSEC_{sum}$ [29]	61.25±0.00	61.46±3.61	41.69±0.00	48.59±0.00	76.80±0.00	85.28±0.00	55.25±0.94	54.45±0.00	38.40±0.77	48.27±0.61
MVEC [27]	37.14±1.37	82.77±0.58	46.33±0.76	67.50±1.27	62.93±2.85	62.93±2.85	45.51±1.08	79.69±0.00	91.40±0.00	55.45±0.00
M^2VEC [28]	52.41±1.52	83.07±0.00	46.77±0.27	77.62±0.00	78.83±3.02	83.52±0.00	44.87±1.10	88.53±0.75	91.20±0.00	56.39±0.21
MMEC (Ours)	64.65±1.32	83.65±0.00	42.44±0.44	80.95±0.00	87.75±4.71	17.26±0.47	53.80±4.35	75.74±0.00	88.60±0.00	63.43±0.00
MEC (Ours)	80.16±1.65	88.44±0.06	76.19±2.84	99.81±0.26	99.40±0.00	95.24±0.00	80.80±4.20	77.51±0.62	99.80±0.00	66.83±9.25

Table 4. Performance of Clustering (NMI%±std%) on Ten Real-world Datasets

Datasets	100leaves	BBC4view	Flowers	MSRC-v1	Mfeat	Notting-Hill	ORL	3Sources	20Newsgroups	Caltech101-7
SC_{bsv} [25]	76.37±0.29	5.80±0.89	32.01±0.53	54.47±1.10	67.96±0.06	65.29±0.00	75.16±0.23	47.14±2.45	45.10±0.66	48.29±2.89
SC_{sum} [25]	66.72±0.86	8.01±2.57	17.99±0.30	39.78±1.20	47.89±0.14	76.07±0.03	69.43±0.29	46.40±4.09	35.16±0.15	40.78±2.23
$rBDLR_{bsv}$ [44]	70.96±0.36	18.77±0.00	37.32±0.32	56.34±0.83	78.49±0.05	1.39±0.00	84.37±0.21	54.60±0.05	25.28±0.16	41.43±0.13
RMSC [37]	67.08±2.17	63.31±0.35	35.32±1.84	67.26±3.28	73.66±2.00	73.30±4.43	85.34±0.02	61.53±0.02	79.20±0.04	49.07±0.01
SEC_{bsv} [21]	78.86±0.54	64.17±0.00	36.66±0.51	63.33±2.95	80.48±1.16	75.77±0.00	35.14±1.90	55.23±1.87	17.35±0.35	53.46±1.62
SEC_{sum} [21]	84.38±0.06	68.00±0.00	43.16±0.14	65.41±0.85	87.17±0.00	91.20±0.00	32.00±1.45	59.21±3.50	86.86±4.47	22.68±1.82
$RSEC_{bsv}$ [29]	52.45±0.00	46.17±0.00	39.09±0.00	56.39±0.00	60.67±0.00	72.04±0.00	49.31±1.01	63.60±0.00	14.80±0.68	60.52±0.72
$RSEC_{sum}$ [29]	42.31±0.00	57.66±0.00	45.19±0.00	43.69±0.00	71.46±0.00	75.39±0.00	41.58±1.23	41.65±0.00	24.45±0.45	43.10±0.65
MVEC [27]	81.80±0.00	69.98±0.12	47.22±0.55	65.13±0.31	61.62±1.54	84.57±1.29	69.47±0.63	75.69±0.00	77.26±0.00	61.28±0.00
M^2VEC [28]	75.43±0.47	66.23±0.00	48.11±1.06	66.54±0.50	68.55±0.00	77.49±0.00	69.64±0.49	77.22±0.54	86.46±0.00	63.18±0.41
MMEC (Ours)	85.77±0.71	67.60±0.00	43.15±0.38	69.41±0.00	83.57±1.20	8.07±0.82	76.90±1.80	67.41±0.00	76.75±0.00	69.07±0.00
TMEC (Ours)	92.81±0.71	80.20±0.05	89.30±1.06	99.57±0.59	98.52±0.00	91.20±0.00	91.26±1.19	71.03±0.00	99.30±0.00	72.89±4.09

- Generally, the proposed TMEC has achieved overwhelming clustering performance in most cases in terms of ACC and NMI values, and the MMEC method has a comparable clustering performance. In detail, our proposed MMEC method shows preferable performance except for the Notting-Hill dataset. Compared with most EC methods, it has higher precision in 100leaves, BBC4view, MSRC-v1, Mfeat, and Caltech101-7 datasets.
- As the best-performed model, TMEC ranks first on nine datasets incldluing 100leaves, BBC4view, Flower, MSRC-v1, Mfeat, Notting-Hill, ORL, 20Newsgroups, and Caltech101-7. In the 3-Sources dataset, TMEC has an ACC 11.02% less than the M^2EC method. However, TMEC has at least improved 5.37% accuracy from the M^2EC method for other datasets. The improvement in the MSRC-v1 dataset is quite remarkable, with 22.19% higher accuracy than the best performance in other ensemble clustering algorithms.
- Compared with other ensemble clustering methods, our proposed TMEC usually has better performance in both ACC and NMI. For example, the proposed TMEC improves the two metrics on Caltech101-7 over M^2VEC by 10.44% and 9.71%, respectively. Therefore, we can conclude that the designed TMEC has substantially enhanced the clustering results over other ensemble clustering methods.
- For the recently proposed multi-view ensemble clustering approaches MVEC and M^2VEC, MVEC is incapable of removing noise and thus performs worse than M^2VEC. However, M^2VEC uses the marginalized denoising process to reduce the disturbance of data corruptions and noises.
- Our TMEC and MMEC methods and RMSC are inspired by the low-rank and sparse decomposition for running efficiency. But, the proposed TMEC has substantial improvements over RMSC due to the benefits of fusing high-level information and exploring high-order correlation among multiple views. Compared with Equation (3), we can conclude that the joint optimization of the low-rank representation Z and the unified indicator matrix H is not always a better way for multi-view ensemble clustering.

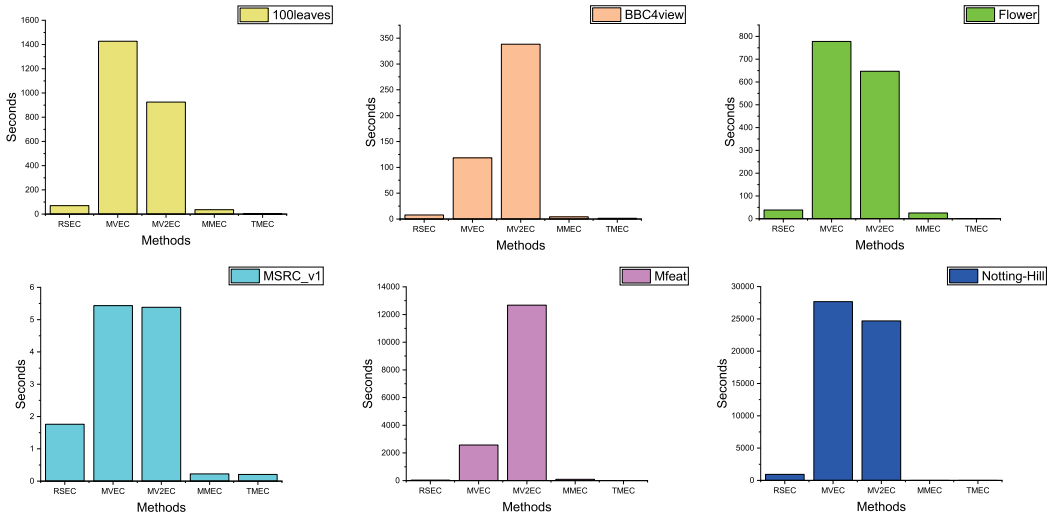

Fig. 2. Time performance on (a) 100Leaves, (b) BBC4views, (c) Flower, (d) MSRC-v1, (e) Mfeat, and (f) Notting-Hill datasets.

(2) *Time Performance:* Another significant advance of our methods is the high running speed. To speed up the algorithms of ensemble clustering, which are based on the self-representation and thus encountered high computational cost, we utilize the scheme of low-rank and sparse decomposition. Figure 2 shows the running times of the proposed methods with several representative ensemble clustering methods, including RSEC, MVEC, M^{2}VEC on 100leaves, BBC4view, Flowers, MSRC-v1, Mfeat, and Notting-Hill datasets. More specifically, we compare the time complexity from the co-association matrices step. As seen from Figure 2, our methods could be faster than other state-of-art ensemble clustering algorithms to a large extent. This is because our proposed methods without matrix inversion operation do not utilize self-representation in the objective function, and low-rank and sparse decomposition has better generalization for low-rank properties over the existing ensemble clustering methods. MMEC and TMEC algorithms usually require several iterations as shown in Figure 4. For most ensemble clustering methods like MVEC and MV^{2}EC, which followed the self-representation framework, although they have promising performance in accuracy and other evaluation indexes, the cost of time is expensive and ineluctable. On the Notting-Hill dataset, our proposed MMEC and TMEC cost around 12.71% and 11.82% time over the third-fastest ensemble clustering method RSEC.

5.3 Model Analysis

In this section, we take the TMEC method as an example and provide an integrated analysis of TMEC concerning parameter analysis and numerical convergence.

(1) *Parameter Analysis:* The proposed TMEC includes one free parameter λ. Parameter λ is to equilibrium the outcome of the low-rank tensor attribute and the noise attribute. We tune it from interval $[0.01 : 0.01 : 0.1]$. In addition, ACC and NMI values of TMEC with diverse λ are shown in Figure 3. The general observation is that TMEC is insensitive to parameter λ. Specifically, on Notting-Hill and Caltech101-7 datasets, all ACC values of TMEC are higher than that of all competing methods, which consistently proves the effectiveness of TMEC. Except for the ORL dataset, ACC values have a slight fluctuation with different parameters on other datasets.

(2) *Convergence Analysis:* To investigate the numerical convergence, we also try to analyze the iteration situation of our TMEC method. Figure 4 shows the values of ACC and NMI, and relative

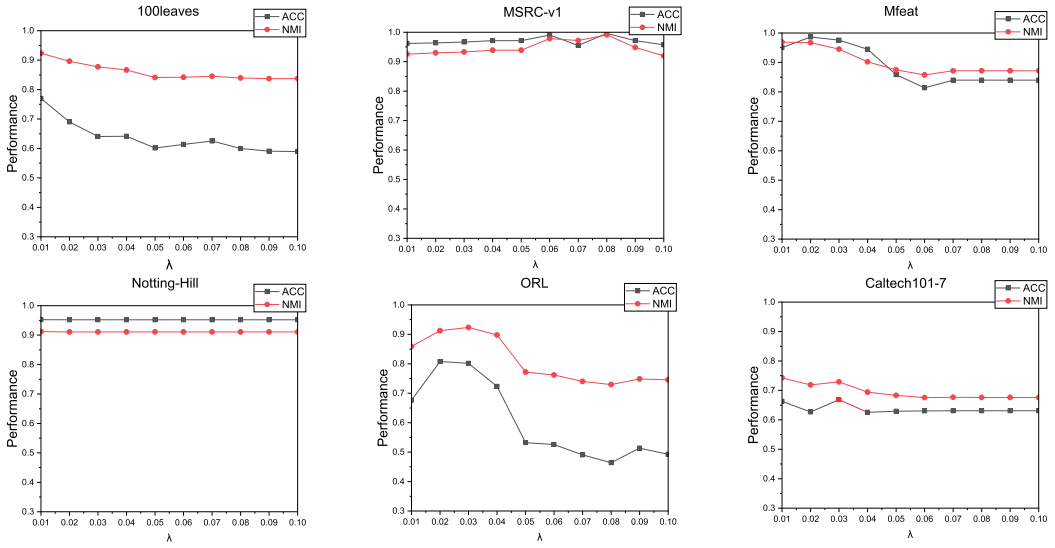

Fig. 3. Parameter tuning with regard to λ on (a)100leaves, (b) MSRC-v1, (c) Mfeat, (d) Notting-Hill, (e) ORL, and (f) Caltech101-7 datasets.

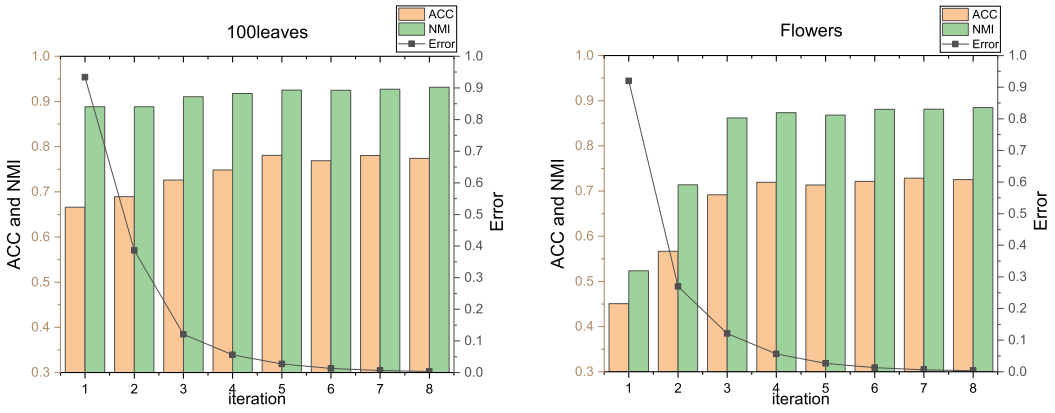

Fig. 4. Relative errors, ACC, and NMI versus iterations on 100leaves and Flowers datasets.

errors with the corresponding iterations on the 100leaves and Flower datasets. It is noted that (1) the values of ACC and NMI have the tendency of being stable as the number of iterations increases, and (2) after about eight iterations, the relative error values become stable. The above phenomena indicate that TMEC converges rapidly, which is also indicated in Figure 2 where the TMEC method has the shortest running time in the ensemble clustering algorithms.

6 CONCLUSION

In this article, we proposed an innovative ensemble clustering scheme focusing on low-rank and sparse decomposition instead of self-representation, thus substantially reducing the computational time and boosting clustering performance. In light of this, two multi-view ensemble clustering methods were developed from the matrix and tensor perspectives, which learned one common low-rank co-association matrix and one low-rank co-association tensor, respectively. Motivated

by the tensor learning method, we split the joint co-association matrices and learn high-level information and high-order correlation from the tensor perspective, such that the designed TMEC has enhanced the clustering performance over our matrix method MMEC. Experimental results have shown that our methods cost fewer operation times to a greater extent than other state-of-the-art multi-view clustering algorithms. In the foreseeable future, there are some unexplored investigations worth pursuing, such as text clustering, image classification, image restoration, and so on.

REFERENCES

[1] Tahani Alqurashi and Wenxin Wang. 2019. Clustering ensemble method. *Int. J. Mach. Learn. Cybernet.* 10 (2019), 1227–1246.

[2] Thierry Bouwmans, Necdet Serhat Aybat, and El-hadi Zahzah. 2016. *Handbook of Robust Low-rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. CRC Press.

[3] Thierry Bouwmans, Namrata Vaswani, Paul Rodriguez, René Vidal, and Zhouchen Lin. 2018. Introduction to the issue on robust subspace learning and tracking: Theory, algorithms, and applications. *IEEE J. Select. Topics in Signal Process.* 12, 6 (2018), 1127–1130.

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* 3, 1 (2011), 1–122.

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

[6] Venkat Chandrasekaran, Pablo A. Parrilo, and Alan S. Willsky. 2010. Latent variable graphical model selection via convex optimization. In *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton'10)*. IEEE, 1610–1613.

[7] Yongyong Chen, Yanwen Guo, Yongli Wang, Dong Wang, Chong Peng, and Guoping He. 2017. Denoising of hyperspectral images using nonconvex low rank matrix approximation. *IEEE Trans. Geosci. Remote Sens.* 55, 9 (2017), 5366–5380.

[8] Yongyong Chen, Shuqin Wang, Chong Peng, Zhongyun Hua, and Yicong Zhou. 2021. Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering. *IEEE Trans. Image Process.* 30 (2021), 4022–4035.

[9] Yongyong Chen, Xiaolin Xiao, and Yicong Zhou. 2019. Low-rank quaternion approximation for color image processing. *IEEE Trans. Image Process.* 29 (2019), 1426–1439.

[10] C.-H. Chou, M.-C. Su, and Eugene Lai. 2004. A new cluster validity measure and its application to image compression. *Pattern Anal. Appl.* 7, 2 (2004), 205–220.

[11] Inderjit S. Dhillon, Subramanyam Mallela, and Rahul Kumar. 2003. A divisive information theoretic feature clustering algorithm for text classification. *J. Mach. Learn. Res.* 3 (2003), 1265–1287.

[12] Xiaoli Zhang Fern and Carla E. Brodley. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the 21st International Conference on Machine Learning*. 36.

[13] Ana L. N. Fred and Anil K. Jain. 2005. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 6 (2005), 835–850.

[14] Danfeng Hong, Wei He, Naoto Yokoya, Jing Yao, Lianru Gao, Liangpei Zhang, Jocelyn Chanussot, and Xiaoxiang Zhu. 2021. Interpretable hyperspectral artificial intelligence: When nonconvex modeling meets hyperspectral remote sensing. *IEEE Geosci. Remote Sens. Mag.* 9, 2 (2021), 52–87.

[15] Yue Hu and Daniel B. Work. 2020. Robust tensor recovery with fiber outliers for traffic events. *ACM Trans. Knowl. Discov. Data* 15, 1 (2020), 1–27.

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

[17] Hongmin Li, Xiucai Ye, Akira Imakura, and Tetsuya Sakurai. 2020. Ensemble learning for spectral clustering. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'20)*. IEEE, 1094–1099.

[18] Lu Li, Wei Li, Qian Du, and Ran Tao. 2020. Low-rank and sparse decomposition with mixture of Gaussian for hyperspectral anomaly detection. *IEEE Trans. Cybernet.* 51, 9 (2020), 4363–4372.

[19] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. 2012. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1 (2012), 171–184.

[20] Hongfu Liu and Yun Fu. 2018. Consensus guided multi-view clustering. *ACM Trans. Knowl. Discov. Data* 12, 4 (2018), 1–21.

[21] Hongfu Liu, Tongliang Liu, Junjie Wu, Dacheng Tao, and Yun Fu. 2015. Spectral ensemble clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 715–724.
[22] Hongfu Liu, Junjie Wu, Tongliang Liu, Dacheng Tao, and Yun Fu. 2017. Spectral ensemble clustering via weighted K-means: Theoretical and practical evidence. *IEEE Trans. Knowl. Data Eng.* 29, 5 (2017), 1129–1143.
[23] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. 2019. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 4 (2019), 925–938.
[24] Shirui Luo, Changqing Zhang, Wei Zhang, and Xiaochun Cao. 2018. Consistent and specific multi-view subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
[25] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*. MIT Press, 849–856.
[26] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* 3 (Dec. 2002), 583–617.
[27] Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. 2017. From ensemble clustering to multi-view clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2843–2849.
[28] Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. 2019. Marginalized multiview ensemble clustering. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 2 (2019), 600–611.
[29] Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. 2019. Robust spectral ensemble clustering via rank minimization. *ACM Trans. Knowl. Discov. Data* 13, 1 (2019), 1–25.
[30] Zhiqiang Tao, Hongfu Liu, Sheng Li, and Yun Fu. 2016. Robust spectral ensemble clustering. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 367–376.
[31] Alexander Topchy, Anil K. Jain, and William Punch. 2003. Combining multiple weak clusterings. In *Proceedings of the 3rd IEEE International Conference on Data Mining*. IEEE, 331–338.
[32] Namrata Vaswani, Thierry Bouwmans, Sajid Javed, and Praneeth Narayanamurthy. 2018. Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery. *IEEE Signal Process. Mag.* 35, 4 (2018), 32–55.
[33] Namrata Vaswani, Yuejie Chi, and Thierry Bouwmans. 2018. Rethinking PCA for modern data sets: Theory, algorithms, and applications [scanning the issue]. *Proc. IEEE* 106, 8 (2018), 1274–1276.
[34] Haishen Wang, Yan Yang, Hongjun Wang, and Dahai Chen. 2013. Soft-voting clustering ensemble. In *Proceedings of the International Workshop on Multiple Classifier Systems*. Springer, 307–318.
[35] Hanrui Wu and Michael K. Ng. 2022. Multiple graphs and low-rank embedding for multi-source heterogeneous domain adaptation. *ACM Trans. Knowl. Discov. Data* 16, 4 (2022), 1–25.
[36] Junjie Wu, Hongfu Liu, Hui Xiong, Jie Cao, and Jian Chen. 2014. K-means-based consensus clustering: A unified view. *IEEE Trans. Knowl. Data Eng.* 27, 1 (2014), 155–169.
[37] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. 2014. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, Vol. 28. 367–376.
[38] Yuan Xie, Bingqian Lin, Yanyun Qu, Cuihua Li, Wensheng Zhang, Lizhuang Ma, Yonggang Wen, and Dacheng Tao. 2020. Joint deep multi-view learning for image clustering. *IEEE Trans. Knowl. Data Eng.* 33, 11 (2020), 3594–3606.
[39] Yuan Xie, Jinyan Liu, Yanyun Qu, Dacheng Tao, Wensheng Zhang, Longquan Dai, and Lizhuang Ma. 2020. Robust kernelized multiview self-representation for subspace clustering. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 2 (2020), 868–881.
[40] Yuan Xie, Dacheng Tao, Wensheng Zhang, Yan Liu, Lei Zhang, and Yanyun Qu. 2018. On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *Int. J. Comput. Vision* 126, 11 (2018), 1157–1179.
[41] Yuan Xie, Wensheng Zhang, Yanyun Qu, Longquan Dai, and Dacheng Tao. 2018. Hyper-Laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning. *IEEE Trans. Cybernet.* 50, 2 (2018), 572–586.
[42] Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. 2017. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7370–7379.
[43] Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. 2018. Generalized latent multi-view subspace clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 1 (2018), 86–99.
[44] Zhao Zhang, Jiahuan Ren, Sheng Li, Richang Hong, Zhengjun Zha, and Meng Wang. 2019. Robust subspace discovery by block-diagonal adaptive locality-constrained representation. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1569–1577.
[45] Di Zhong, HongJiang Zhang, and Shih-Fu Chang. 1996. Clustering methods for video browsing and annotation. In *Storage and Retrieval for Still Image and Video Databases IV*, Vol. 2670. SPIE, 239–246.
[46] Jie Zhou, Hongchan Zheng, and Lulu Pan. 2019. Ensemble clustering based on dense representation. *Neurocomputing* 357 (2019), 66–76.

Received 29 August 2022; revised 27 January 2023; accepted 26 March 2023