Multi-View Partitioning via Tensor Methods

Xinhai Liu, Shuiwang Ji, Wolfgang Glänzel, and Bart De Moor, Fellow, IEEE

Abstract—Clustering by integrating multi-view representations has become a crucial issue for knowledge discovery in heterogeneous environments. However, most prior approaches assume that the multiple representations share the same dimension, limiting their applicability to homogeneous environments. In this paper, we present a novel tensor-based framework for integrating heterogeneous multi-view data in the context of spectral clustering. Our framework includes two novel formulations; that is, multi-view clustering based on the integration of the Frobenius-norm objective function (MC-FR-01) and that based on matrix integration in the Frobenius-norm objective function (MC-FRMI). We show that the solutions for both formulations can be computed by tensor decompositions. We evaluated our methods on synthetic data and two real-world data sets in comparison with baseline methods. Experimental results demonstrate that the proposed formulations are effective in integrating multi-view data in heterogeneous environments.

Index Terms—Multi-view clustering, tensor decomposition, spectral clustering, multi-linear singular value decomposition, higher-order orthogonal iteration

I. INTRODUCTION

In many real-world scenarios, each object can be described by multiple sets of features. For example, in scientific literature mining, both the textual content and the citation link between articles are often used in the knowledge discovery processes [25]. In multiplex network analysis, we are given a set of multiple networks that share the same set of nodes but possess network-specific links representing different types of relationships between nodes [29]. A particular instance of this scenario is the social network of university students, which may include symmetrized connections from (i) Facebook friendship, (ii) picture friendship, (iii) roommate relations, and (iv) student housing-group preference. These diverse individual activities result in multiple relationship networks among students. Such a learning scenario is called multi-view learning, since each feature set describes a relationship among them.

Multi-view clustering refers to the clustering of the same set of objects with multi-view features, either from various information sources or from different feature representations. Compared with the clustering that is implemented on single-view data, multi-view clustering is expected to yield robust and novel partition results by exploiting the complementary information in different views. One of the recent developments in clustering is the spectral clustering technique, which has seen an explosive proliferation over the past several years [44]. Among many other factors, such as easy implementation and efficiency, one of the key advantages of spectral clustering is that it is based on the relaxation of a global clustering criterion (i.e., normalized cuts). Spectral clustering has been widely employed in many real applications, from image segmentation to community detection. Although spectral clustering [28] works well on single-view data, it is not well suited for the clustering of multi-view data, since it is inherently based on matrix decompositions.

Recently, several multi-view clustering algorithms have been proposed [1], [3], [5], [25], [26], [37], [40], [47]. These multi-view clustering techniques have been shown to yield better performance in comparison to single-view techniques. However, prior methods have some limitations that prevent their wide applicabilities, as we will discuss in the related work. For instance, some techniques assume that the dimensions of the features in multiple views are the same, limiting their applicability to the homogeneous settings. Some other techniques only concentrate on the clustering of two-view data so that it might be hard to extend them to more than a two-view situation [3]. In addition, an appropriate weighting scheme is lacking for these multiple views although coordinating various information from them is also one crucial step in gaining good clustering results [37], [41]. A unified framework that can integrate various types of multi-view data is lacking to date [26], [40].

Tensors are higher-order generalizations of matrices. They have been successfully applied to several domains, such as chemometrics, signal processing, Web search, data mining, scientific computing and image recognition [10], [21], [22], [34], [38], [45]. Traditionally, tensor-based methods have been used to model multi-view data [21], and tensor methods are very powerful tools to analyze the latent pattern hidden in multi-view data. Tensor decompositions capture multi-linear structures in higher-order data-sets, where the data have more than two modes. Tensor decompositions and multi-way analysis allow for extracting hidden (latent) components (cluster structure) and investigating complex relationship among them.

In this paper, we propose a multi-view clustering framework based on tensor methods. Our formulations model the multi-view data as a tensor and seek a joint latent optimal subspace by tensor analysis. Our framework can leverage the inherent consistency among multi-view data and integrate their information seamlessly. Apart from other multi-view clustering strategies, which are usually devised for ad hoc application, our method provides a general framework in which some limitations of prior methods are overcome systematically. In particular, our framework can be extended to various types of multi-view data. Almost any multiple similarity matrices of the same entities are
allowed to be embedded into our framework. In addition, since our framework can obtain a joint optimal subspace, it can be easily extended to other related machine learning tasks, such as classification, spectral embedding and collaborative filtering. Our framework consists of two novel algorithms: multi-view clustering based on optimization integration of the Frobenius-norm objective function (MC-FR-OI) and that based on matrix integration in the Frobenius-norm objective function (MC-FR-MI). In particular, MC-FR-MI can assign each view a suitable weight to boost the clustering. For each strategy, we provide the relevant tensor based solutions. Similar to other variants of PCA in machine learning applications [46], our strategy can be considered as a multi-view PCA analysis.

Figure 1 illustrates the potential benefit of multi-view clustering. The figure shows two groups of data points in a 3-D space. Suppose that due to limitations of the measurement system (such as 2-D cameras in the real world), only 2-D projections of the data points can be observed (such as, X-Y projection, Y-Z projection and X-Z projection in a 3-D X-Y-Z coordinate system). Each of the three projections yields what we call a single-view data set. The figure shows that separation of the two clusters is not possible from any of the three projections separately. However, the three views together do contain the information that was present in the original data. Combination of the three views does not automatically allow proper clustering. The middle right part of the figure shows the result of spectral projection by means of multiple kernel fusion (MKF). MKF does not yield satisfactory results here. In this paper we present a new class of algorithms for multi-view partitioning. The lower right part of Figure 1 shows the results obtained by our MC-OI-MLSDV algorithm. The latent cluster structure hidden amid the multi-view data has clearly been recovered here.

To the best of our knowledge, our work is the first unified attempt to address multi-view clustering within the framework of tensor methods. The key contributions of our work can be summarized as follows:

- We propose to model multi-view data as a tensor and develop a new framework of multi-view clustering by tensor methods.
- We present two novel multi-view clustering strategies with their tensor solutions.
- We systematically evaluate our methods on both a synthetic data set and two real applications.

The rest of the paper is organized as follows. To start, Section II reviews the related work. Then, Section III introduces the concepts of spectral clustering. Next, Section IV presents our tensor based multi-view clustering algorithms. After that, Section V demonstrates the experimental results on synthetic data and practical applications. The related research issues are discussed in Section VI. Finally, we conclude in Section VII.

**Notation:** To facilitate the distinction between scalars, vectors, matrices, and higher-order tensors, the type of a given quantity will be reflected by its representation: scalars are denoted by lower-case letters \((a, b, \ldots; \alpha, \beta, \ldots)\), vectors are written as italic capitals \((A, B, \ldots)\), matrices correspond to boldface capitals \((\mathbf{A}, \mathbf{B}, \ldots)\), and tensors are written as calligraphic letters \((\mathcal{A}, \mathcal{B}, \ldots)\). This notation is consistently used for lower-order parts of a given quantity. For instance, \(a_{ij}\), \(a_{ij}\), and \(a_{ijk}\) denote an entry of a vector \(a\), a matrix \(A\) and a tensor \(\mathcal{A}\), respectively. The Kronecker product is denoted by \(\otimes\). For \(A \in \mathbb{R}^{T \times J}\), \(\text{vec}(A) = (a_{11} \ a_{21} \ldots \ a_{IJ})^T \in \mathbb{R}^{IJ}\) is the vector in which the columns of \(A\) are stacked on top of each other. diag() is the column vector that is given by the diagonal of its matrix argument.

**II. Related work**

**A. Multi-view clustering**

Bickel and Scheffere [3] propose a multi-view clustering method that extends k-means and hierarchical clustering to deal with data with two conditionally independent views. A multi-view clustering strategy via canonical correlation analysis (CCA) is presented in [5]. This method assumes that the views are uncorrelated given the cluster label. The above algorithms only concentrate on the clustering of two-view data thus it might be hard to extend them to more than two-view situations. Meanwhile our strategy is applicable to any multi-view situation. Long et al. [26] formulate a multi-view spectral clustering method while investigating multiple spectral dimension reduction. A clustering method based on linked matrix factorization is introduced to fuse information from multiple graphs in [41]. Zhou et al. [47] develop a multi-view clustering strategy via generalizing the normalized cut from a single view to multiple views and subsequently they build a multi-view transductive inference. In the above algorithms, a common problem is that the analysis of inherent relationship among multi-view data might be neglected. While in our tensor based strategy, the multi-linear relationship among multi-view data is taken into account. Furthermore, Long et al. propose a general model based on collective factorization of the related matrices for clustering multi-type relational data [27]. The strategy focuses on the clustering of multi-type interrelated data objects, rather than on the clustering of the same objects using multiple representations as in our research.

**B. Community detection of multi-view networks**

Tang et al. propose the concept of feature integration to implement the clustering of multi-view social networks [40]. Based on modularity optimization, Mucha et al. [29] develop a generalized framework of network quality functions that allow studies of community structure in a general setting encompassing networks that evolve over time, have multiple types of links (multiplexity), and have multiple scales. These methods are applicable to specific type of data with sparse links while our strategy is devised for general data.

**C. Kernel fusion and clustering ensemble**

Multiple kernel learning aims at finding a combination of kernels to optimize for classification or clustering [20], [25]. Such a solution might sound natural, but its underlying principal is not clear [47]. In addition, the heavy computation of their convex optimization makes them only applicable to small databases [25]. Meanwhile, with the recent research progress in tensor decomposition [32], our strategy has the potential to tackle large-scale databases. Clustering ensemble is also known as clustering aggregation or consensus clustering, which integrates different partitions into a consolidated partition with a consensus function [1], [37]. However, clustering ensemble methods usually concentrate on single-view data to overcome the drawback of k-means. In fact, clustering ensemble is embedded into our strategy to facilitate the final partition.
D. Tensor based clustering

Sun et al. [38] introduce a dynamic tensor analysis (DTA) algorithm and its variants, and apply them to anomaly detection and multi-way latent semantic indexing. It seems their clustering method is designed for dynamic stream data. Dunlavy et al. [10] apply PARAFAC decomposition for analyzing scientific publication data with multiple linkage. Selee et al. create a new tensor decomposition called Implicit Slice Canonical Decomposition (IMSCAND) to group information when multiple similarities are known [34]. The last two ideas that integrate multi-view data as a tensor are similar to ours. But our methods rely on a Tucker-type tensor decomposition. Furthermore, in these methods, all single-view data are considered equally important, while we will present a technique that compute weights for the different views.

III. SPECTRAL CLUSTERING

Spectral clustering was originally derived based on relaxation of the normalized cut formulation for clustering [35]. In particular, spectral clustering involves a matrix trace optimization problem [28], [30]. We show in this paper that the spectral clustering formalism can be extended to deal with multi-view problems based on tensor computations.

A. Single-view spectral clustering

We first consider spectral clustering in the single-view setting. Suppose $Y \in \mathbb{R}^{N \times M}$ is the relaxed assignment matrix, where $N$ in which the vertices $V$ represent the data points and the edges $e_{ij} \in E$ characterize the similarity between data points quantified by $s_{ij}$. Usually, the similarity measure is symmetric, and the graph is undirected. The affinity matrix of the graph $G$ is the matrix $S$ with entry in row $i$ and column $j$ equal to $s_{ij}$. The degree of the vertex $v_i$, defined as

$$d_i = \sum_{j=1}^{N} s_{ij},$$

is the sum of all the weights of edges connected to $v_i$. The degree matrix $D$ is a diagonal matrix containing the vertex degrees $d_1, \ldots, d_N$ on the diagonal. It follows from the spectral embedding formalism [28], [30], [35] that the Laplacian matrix is defined as $L = D - S$, and the normalized Laplacian matrix, corresponding to the normalized cuts (Ncut), is defined as

$$L_{\text{Ncut}} = D^{-1/2}LD^{-1/2} = I - S_N,$$

where $S_N$ is the normalized similarity matrix and defined as

$$S_N = D^{-1/2}SD^{-1/2}.$$

The matrices $S_N$ and $L_{\text{Ncut}}$ have the same eigenvectors, and their eigenvalues are related as $\lambda^{(S_N)} = 1 - \lambda^{(L_{\text{Ncut}})}$, where $\lambda^{(S_N)}$ and $\lambda^{(L_{\text{Ncut}})}$ are the eigenvalues for $S_N$ and $L_{\text{Ncut}}$, respectively.
is the number of data points and $M$ is the number of clusters. The spectral clustering problem can be expressed as

$$
\min \text{trace}(U^T L_{Ncut} U), \\
\text{s.t. } U^T U = I.
$$

(4)

It follows from the Ky Fan theorem [31] that the optimal solution to the optimization problem in (4) is given by the $M$ dominant eigenvectors of $L_{Ncut}$. Considering the relationship between $S_N$ and $L_{Ncut}$, spectral clustering can equivalently be formulated as

$$
\max \text{trace}(U^T S_N U), \\
\text{s.t. } U^T U = I.
$$

(5)

Since $S_N$ is positive semi-definite, spectral clustering can also be formulated as the following Frobenius norm optimization problem:

$$
\max_U \|U^T S_N U\|_F^2, \\
\text{s.t. } U^T U = I.
$$

(6)

The objective functions in (5) and (6) are different, but they have the same solution, namely, the columns of the optimal matrix $U$ span the dominant eigenspace of $S_N$.

B. Multi-view spectral clustering

We propose different strategies for the integration of multi-view data in the context of spectral clustering.

1) Multi-view clustering by trace maximization (MC-TR-I): The first strategy is to add objective functions of the type in (5), associated with the different views. We consider:

$$
\max_U \sum_{k=1}^K \text{trace}(U^T S_N^{(k)} U) = \text{trace}(U^T (\sum_{k=1}^K S_N^{(k)}) U), \\
\text{s.t. } U^T U = I,
$$

(7)

where $S_N^{(k)}$ is the normalized similarity matrix for the $k$th view and $U$ is the common factor shared by the views. This corresponds to Multiple Kernel Fusion (MKF) with a linear kernel [20], see Section V-A.

As an alternative, we may optimize a weighted combination of objective functions, where the weights are learnt from the data:

$$
\max_{U,W} \sum_{k=1}^K w_k \text{trace}(U^T S_N^{(k)} U) = \max_{U,W} \text{trace}(U^T (\sum_{k=1}^K w_k S_N^{(k)}) U), \\
\text{s.t. } U^T U = I, \quad W \geq 0 \text{ and } \|W\|_F = 1.
$$

(8)

2) Multi-view clustering by integration of the Frobenius-norm objective function (MC-FR-OI): Note that all terms in the objective function $\sum_{k=1}^K \sum_{m=1}^M (U^T S_N^{(k)} U)_{m,m}$ in (7) are nonnegative, since $S_N^{(k)}$ is positive (semi)definite, $1 \leq k \leq K$. Instead, we might consider the optimization of $\sum_{k=1}^K \sum_{m=1}^M (U^T S_N^{(k)} U)_{m,m}^2$. This corresponds to adding objective functions of the type in (6):

$$
\max_U \sum_{k=1}^K \|U^T S_N^{(k)} U\|_F^2, \\
\text{s.t. } U^T U = I.
$$

(9)

IV. MULTI-VIEW SPECTRAL CLUSTERING VIA TENSOR METHODS

Following the two multi-view clustering strategies discussed above, we present the tensor-based solutions in this Section. Compared to the single-view spectral clustering, which is solved by matrix decomposition, we formulate our multi-view clustering by tensor decomposition. The overview of the tensor-based method is depicted in Figure 2. As shown in the left part of Figure 2, the goal of single-view spectral clustering is to find an optimal latent subspace from single-view data. In contrast, with multi-view data, we want to obtain a joint optimal subspace with the aid of tensor methods.

A. Background on tensors

In this section we provide some basic background on tensors and low multilinear rank approximation. We refer to [6]–[8], [22], [24], [36] for more details. A tensor is a multi-way array. The order of a tensor is the number of modes (or ways). A first-order tensor is a vector, a second-order tensor is a matrix and a tensor of order three or higher is called a higher-order tensor. We only discuss third-order tensor methods that are relevant to our problem.

Matrix unfolding is the process of re-ordering the elements of a tensor into a matrix. The mode-1, mode-2 and mode-3 matrix unfoldings of a tensor $A \in \mathbb{R}^{I \times J \times K}$ are denoted by $A_{(1)}$, $A_{(2)}$ and $A_{(3)}$, respectively. The definition follows from Figure 3.

A tensor can be multiplied by a matrix as follows. Consider matrices $B \in \mathbb{R}^{I \times J}$, $C \in \mathbb{R}^{J \times J}$ and $D \in \mathbb{R}^{K \times K}$, then the mode-1 product $A_{\times 1} B$, mode-2 product $A_{\times 2} C$ and mode-3


Fig. 2. Comparison between single view (left) and multi-view (right) spectral clustering.

The product $A \times_3 D$ are defined by

\[
(A \times_1 B)_{i,j,k} = \sum_{i=1}^{I} a_{ij} b_{i,k}, \quad \forall i,j,k,
\]

\[
(A \times_2 C)_{i,j,k} = \sum_{j=1}^{J} a_{i,j} c_{j,k}, \quad \forall i,j,k,
\]

\[
(A \times_3 D)_{i,j,k} = \sum_{k=1}^{K} a_{i,j} d_{k,k}, \quad \forall i,j,k,
\]

respectively. The Frobenius norm of $A$ is defined by

\[
\|A\|_F = \left( \sum_{i,j,k} a_{i,j,k}^2 \right)^{1/2}.
\]

Multilinear singular value decomposition (MLSVD) is one of the possible higher-order extensions of matrix singular value decomposition (SVD) [7], [42], [43]. It decomposes $A$ as

\[
A = B \times_1 U \times_2 V \times_3 W,
\]

in which now $U \in \mathbb{R}^{I \times R_1}$, $V \in \mathbb{R}^{J \times R_2}$ and $W \in \mathbb{R}^{K \times R_3}$ are column-wise orthonormal with $R_1 \leq I$, $R_2 \leq J$, $R_3 \leq K$, and in which $B \in \mathbb{R}^{R_1 \times R_2 \times R_3}$. The triplet $(R_1, R_2, R_3)$ is the trilinear rank of the approximand and (12) is a case of what is known as low multilinear rank approximation. It can be shown that the minimization problem is equivalent with the following maximization problem [8], [23]:

\[
\max_{U,V,W} \|A \times_1 U^T \times_2 V^T \times_3 W^T\|_F^2.
\]  

(13)

Analogous to low-rank matrix approximation, one may consider truncated MLSVD for solving (12)–(13), i.e., one may take the columns of $U,V,W$ in (12)–(13) equal to the dominant multilinear singular vectors of $A$. Contrary to the matrix case, the approximation is not optimal in general. However, the result is often fairly good and MLSVD truncation is easy to implement. While in the matrix case the sum of the squared discarded singular values give the approximation error, in the tensor case the discarded multilinear singular values yield an upper bound on it [7].

There exist a number of algorithms for the actual optimization in (12)–(13). The most popular technique is the higher-order orthogonal iteration (HOOI), which is an algorithm of the alternating least-squares (ALS) type [8], [23]. In each iteration step, the estimate of one of the matrices $U,V,W$ is optimized, while the other two are kept fixed. It follows from

\[
\|A \times_1 U^T \times_2 V^T \times_3 W^T\|_F^2 = \|U^T (A_{(1)}(V \otimes W))\|_F^2
\]

(14)

that the optimal $U$, given $V$ and $W$, is determined by the $R_1$-dimensional dominant subspace of the column space of $A_{(1)}(V \otimes W)$. The optimization with respect to $V$ and $W$ is analogous. In practice the convergence is observed to be linear, with a convergence coefficient that is larger as the problem is better conditioned in the sense of [12]. Alternative algorithms are the
trust region method based on truncated conjugate gradient in [18],
the quasi-Newton algorithms in [33] and the Newton algorithms
in [11], [19]. Truncated MLSVD is often used as initial value.
Numerical experiments in [17] suggest that, if there is a gap
between the \( R_n \)th and the \((R_n + 1)\)th mode-\( n \) singular values,
\( n = 1, 2, 3 \), one can expect algorithms to find the global optimum.
In the same paper it is proved that, if there is a gap and there
are nevertheless several local optima, then these are close, both
in terms of the cost function value and in terms of the matrices \( U, V \)
and \( W \). The absence of a gap may indicate the presence of several
local optima for which the cost function value is close. Recent
research includes the generalization of numerical algorithms for
low-rank approximation of large matrices to low multilinear rank
approximation of large higher-order tensors [32].

B. Tensor construction

There are several options for constructing a tensor from multi-
view data. In [15], a tensor is constructed by stacking the object-
by-feature matrices derived from multiple views in a tensor as
shown in the left part of Figure 4. This construction is only
applicable to the scenario of homogeneous data sources, where the
dimensions of different feature spaces are the same. In fact, many
multi-view applications deal with heterogeneous data sources in
which the dimensions of various feature spaces are different.
For instance, in the application to scientific publication analysis
in Section V-D, the dimension of the citation feature space is
8,305 while the dimension of the text feature space is more than
600,000.

Consequently, in this paper we make a construction that is
independent of data dimension, thereby enabling the integration
of heterogeneous data sources. We will work with the similarity
tensor \( A \in \mathbb{R}^{N \times N \times K} \) obtained by stacking the similarity
matrices \( S^{(1)}_N, S^{(2)}_N, \ldots, S^{(K)}_N \) associated with the different views.
The construction of the similarity tensor is illustrated in the right
part of Figure 4. Since the similarity of each view is computed in
a different space, normalization is required. In this respect,
our definition of similarity matrix in (3) may be regarded as a
normalization step.

C. MC-TR-I

The column space of the optimal matrix \( U \) in (7) is the domi-
nant eigenspace of \( \sum_{k=1}^{K} S_k^{(k)} \). The pseudo-code is as follows.

Algorithm IV.1: MC-TR-I-EVD \( (S^{(1)}_N, S^{(2)}_N, \ldots, S^{(K)}_N, M) \)

\textbf{comment:} \( M \) is the number of clusters
\textbf{step 1.} Build a combined similarity matrix \( \sum_{k=1}^{K} S_k^{(k)} \)
\textbf{step 2.} Obtain \( U \) by eigenvalue decomposition
\textbf{step 3.} Normalize the rows of \( U \) to unit length
\textbf{step 4.} Calculate the cluster \( idx \) with k-means on \( U \)
\textbf{return} \( (idx : \text{the clustering label}) \)

The problem in (8) can be written as:
\[
\max_{P(U)} P(U) \cdot W, \quad \text{s.t. } U^T U = I \quad \text{and} \quad \|W\|_F = 1,
\]
where \( P(U) = (\text{trace}(U^T S^{(1)}_N U) \ldots \text{trace}(U^T S^{(K)}_N U)) \). Note
that, compared to (8), the nonnegativity constraint on \( W \) has been
dropped in (15). Since \( S_k^{(k)} \) is positive (semidefinite, \( 1 \leq k \leq K, \)
all entries of \( P(U) \) are nonnegative. Given \( U \), the optimal \( W \) is
just \( P(U) \) scaled to unit-norm, and hence satisfies automatically
the nonnegativity constraint. The overall solution can be computed in
an alternating fashion by additionally deriving from (8) that the
optimal \( U \), given \( W \), follows from the dominant eigenspace
of \( \sum_{k=1}^{K} w_k S^{(k)}_N \). The computation of \( P(U) \) requires \( O(2N^2 K) \)
flops, the construction of \( \sum_{k=1}^{K} w_k S^{(k)}_N \) also requires \( O(2N^2 K) \)
flops and the computation of its eigenspace \( O(bNM^2) \) flops. The pseudo-code is as follows.

Algorithm IV.2: MC-TR-I-EVD \( (S^{(1)}_N, S^{(2)}_N, \ldots, S^{(K)}_N, M) \)

\textbf{step 1.} Initialize e.g. by MC-TR-I-EVD
\textbf{while} <\text{convergence}>
\textbf{iteration step 2.1.} Obtain \( P(U) \)
\textbf{iteration step 2.2.} Calculate the weighting vector \( W \)
by scaling \( P(U) \) to unit-norm
\textbf{iteration step 2.3.} Obtain the relaxed assignment matrix \( U \)
from the dominant eigenspace of \( \sum_{k=1}^{K} w_k S^{(k)}_N \)
\textbf{step 3.} Normalize the rows of \( U \) to unit length
\textbf{step 4.} Calculate the cluster \( idx \) with k-means on \( U \)
\textbf{return} \( (idx : \text{the clustering label}) \)

D. MC-TR-OI

We first discuss the objective function integration approach for
multi-view clustering. The problem in (9) can be written as
\[
\max_U \| A_{\times 1} U^T \times_2 U^T \times_3 I \|^2_F,
\]
in which \( U \in \mathbb{R}^{N \times M} \) has orthonormal columns. If we take into
account the equivalence between (12) and (13), the problem can be
visualized as in Figure 5.

As explained in Section IV-A, an approximate solution to (16)
can be obtained from truncated MLSVD. Here, \( U \) is determined
by the \( M \) dominant mode-1 singular vectors of \( A \), i.e., it follows from the \( M \) dominant left singular vectors of \( A(1) \). Because of the partial symmetry of \( A \), \( A(2) \) yields the same \( U \). We call this method MC-FR-OI-MLSV. Although the approximation is not optimal, the results are often quite good and the algorithm is easy to implement. The computational cost is low, namely \( O(6NM^2) \) flops. The pseudo-code of MC-FR-OI-MLSV is as follows:

**Algorithm IV.3:** MC-FR-OI-MLSV (\( S^{(1)}, S^{(2)}, \ldots, S^{(K)}, M \))

**comment:** \( M \) is the number of clusters

- step 1. Build a similarity tensor \( A \)
- step 2. Obtain the unfolding matrices \( A(1), A(2) \) and \( A(3) \)
- step 3. Obtain an initial \( U_0 \) and \( V_0 \) by MLSD
- while \(<\text{convergence}>\)
  - iteration step 4.1. \( U_{i+1} \) in dominant subspace of \( A(1)(V_0 \otimes I) \)
  - iteration step 4.2. \( V_{i+1} \) in dominant subspace of \( A(2)(U_0 \otimes I) \)
- comment: \( i \) is the counter of iteration
- step 5. Normalize the rows of \( U \) to unit length
- step 6. Calculate the cluster idxs with k-means on \( U \)
- return (idxs: the clustering label)

Both MC-FR-OI-MLSV and MC-FR-OI-HOOI imply a joint matrix compression, as shown in Figure 6. In the case of low multilinear rank approximation, the \((M \times M)\) frontal slices of the core tensor are not necessarily diagonal.

**E. MC-FR-MI**

The problem in (10) can be written as

\[
\max_{U,W} \|A \times_1 U^T \times_2 U^T \times_3 W^T\|_F^2,
\]

s.t. \( U^T U = I \), \( \|W\|_F^2 = 1 \).

(17)

Note that, compared to (10), the nonnegativity constraint on \( W \) has been dropped in (17). Since \( S_N^{(k)} \) is positive (semi)definite, \( U^T S_N^{(k)} U \) is positive (semi)definite, \( 1 \leq k \leq K \). Theorem ?? in the Supplementary material 6 now implies that, for any \( U \), the entries of the optimal \( W \) have the same sign. Since the value of the objective function in (17) is not affected by the sign of \( W \), we can assume that all the weights are nonnegative.

If we take into account the equivalence between (12) and (13), the problem can be visualized as in Figure 7. The matrix \( U \) represents the optimal subspace while the vector \( W \) yields the weights of the different views.

**Algorithm IV.4:** MC-FR-OI-HOOI (\( S^{(1)}, S^{(2)}, \ldots, S^{(K)}, M \))

- step 1. Build a similarity tensor \( A \)
- step 2. Obtain the unfolding matrices \( A(1), A(2) \) and \( A(3) \)
- step 3. Obtain an initial \( U_0 \) and \( V_0 \) by MLSD
- while \(<\text{convergence}>\)
  - iteration step 4.1. \( U_{i+1} \) in dominant subspace of \( A(1)(V_0 \otimes I) \)
  - iteration step 4.2. \( V_{i+1} \) in dominant subspace of \( A(2)(U_0 \otimes I) \)
- comment: \( i \) is the counter of iteration
- step 5. Normalize the rows of \( U \) to unit length
- step 6. Calculate the cluster idxs with k-means on \( U \)
- return (idxs: the clustering label)

An equivalent but more efficient implementation is obtained by taking into account that \( W \) is not a matrix but a vector. The pseudo code is given as Algorithm IV.6. The matrix \( U_{i+1} \) in step 4.1 of Algorithm IV.5 is just equal to the product \((\sum_{k=1}^{K} S^{(k)}(W_{i+1})_k)V_i\). Like-wise, the matrix \( V_{i+1} \) in step 4.2 is equal to \((\sum_{k=1}^{K} S^{(k)}(W_{i+1})_k)U_{i+1}\). Alternating until convergence between steps 4.1 and 4.2 of Algorithm IV.5 yields the same matrix for \( U \) and \( V \). The scheme is known as the Orthogonal Iteration for the computation of the dominant eigenspace of
Algorithm IV.6: MC-FR-MI-HOOI ($S^{(1)}$, $S^{(2)}$, ..., $S^{(K)}$, $M$)

step 1. Build a similarity tensor $A$.
step 2. Obtain the unfolding matrices $A_{(1)}$, $A_{(2)}$ and $A_{(3)}$.
step 3. Obtain an initial $U_0$ by MLSVD.
while $<\text{convergence} >$
  do iteration step 4.1. Calculate $W_{i+1}$ as the dominant left singular vector of $A_{(3)}(U_i \otimes U_i)$.
  iteration step 4.2. Compute a new integration matrix $\tilde{S}$ as $\sum_{i}^{K} W_{i+1}(S^{(i)})$.
  iteration step 4.3. Obtain $U_{i+1}$ by eigenvalue decomposition of $\tilde{S}$.
  comment: $i$ is the counter of iteration.
step 4. Calculate the cluster $idx$ with $k$-means on $U$.
step 5. Normalize the rows of $U$ to unit length.
step 6. Calculate the cluster $idx$ with $k$-means on $U$.
return ($idx$ : the clustering label).

Remark 1: In the MC-OI framework we discussed two variants, namely MC-FR-OI-MLSVD and MC-FR-OI-HOOI. In the MC-MI framework we have only discussed MC-FR-MI-HOOI. The reason is that tests indicated that here mere truncation of the MLSVD, in which the third mode only one vector is retained, often yields results that are not satisfactory.

V. EXPERIMENTAL EVALUATION

In this Section, we report experimental results of the proposed multi-view partition strategies in comparison with baseline multi-view clustering methods.

A. Baseline methods

We compare with the following six baseline methods.

- Multiple kernel fusion (MKF): Joachims et al. [20] integrate different kernels by linear combination for hybrid clustering. The similarity matrix defined in (3) can be regarded as a linear kernel as well. The clustering result of MKF is equal to our MC-TR-I-EVD since the MC-TR-I-EVD is actually the average combination of multiple similarity matrices, so we combine them for the comparison.

- Feature integration (FI) [40]: With the spectral analysis of each view, their structure features are extracted and then integrated, and SVD is then implemented to obtain the cross-view principal components for clustering.

- Strehl’s clustering ensemble algorithm (SA) [37]: Strehl & Ghosh formulate the optimal consensus as the final partition that shares the most information with the partitions of all single-view data to combine. Three heuristic consensus algorithms (cluster-based similarity partition algorithm [CSPA], hyper-graph partition algorithm [HGPA] and meta-clustering algorithm [MCLA]) based on graph partitioning are employed to obtain the combined partition. In this work, the ensemble consists of single view from each view. Due to the low computational costs of these techniques, it is quite feasible to use a supra-consensus function that evaluates all three approaches against the objective function and picks the best solution for a given situation [37]. Therefore which exact heuristic consensus algorithm is adopted relies on each data. In our experiments, MCLA is adopted for all three data sets since it obtains the largest ANMI value for each data respectively. The code of SA is available by the authors.

- AdacVote [1]: Ayad & Kamel propose a cumulative vote weighting method (AdacVote) to compute an empirical probability distribution summarizing the clustering ensemble.

- CP-ALS [4], [14]: The CANDECOMP/PARAFAC (CP) decomposition is usually solved by an alternating least squares (ALS) algorithm, for which we use a tensor toolbox for MATLAB [2]. We adopt the default initialization and parameter setting as defined in the toolbox itself.

- Linked matrix factorization (LMF): In Tang’s work [41], a quasi-Newton method named Limited memory BFGS (L-BFGS) is adopted for the optimization of LMF. We implement this algorithm with the aid of an optimization toolbox for MATLAB named Poblano [9]. Since LMF is sensitive to initialization, we initialize it by MLSVD that usually provides a good initialization. In addition, the optimization parameters are set as the default setting of the toolbox.

Furthermore, we initialize both MC-FR-OI-HOOI and MC-FR-MI-HOOI by truncated MLSVD. We initialize MC-TR-EVDt with the result of MC-TR-I-EVD (MKF).

B. Performance measures

Regarding clustering evaluation, the data sets used in our experiments are provided with labels. Therefore the clustering performance is evaluated comparing the automatic partitions with the labels using Adaptive Rand Index (ARI) [16] and Normalized Mutual Information (NMI) [37]. To evaluate the ARI and NMI performance, we set the number of clusters for journal data to $M = 7$ and $M = 14$ for disease data.

In order to overcome the drawback of the $k$-means algorithm which is sensitive to various initializations, we adopt the combination of clustering ensemble of SA method and $k$-means for both spectral clustering and multi-view clustering. In particular, we first repeat each clustering method 50 times and use the SA method on the clustering ensemble to obtain the final consensus partition. Consequently, the final partition obtained by each clustering algorithm is unique.

C. Experiment on a synthetic multiplex network

We first evaluate and compare different clustering strategies applied to the synthetic multi-view data. The synthetic data has three communities (clusters), which have 50, 100 and 200 members respectively [39]. We generate various views of interactions among these 350 vertices, that is, each view forms a network that shares the same vertices but has a different interaction pattern. For each view, group members connect with each other following a randomly generated within-group interaction probability. The interaction probability differs with respect to groups at distinct views. After that, we add some noise to the network by randomly connecting any two vertices with low probability. The different views demonstrate different interaction patterns. In this multi-view network that is called a multiplex network according to [29],

we construct four interaction matrices, each of whose elements is the interaction strength of a pair of vertices. The visualization of the four adjacent matrices is shown as Figure 8.

Fig. 8. Visualization of the adjacent matrices of a synthetic multiplex network.

In Table I, we list the clustering evaluations of spectral clustering for each single-view data as well as those of multi-view clustering methods. First, it is clear that most multi-view clustering results are better than single-view clustering results. This could be easily explained by the patterns shown in Figure 8. The first view of the network (left above) only shows one group, and the fourth view (right below) involves another group with the other two groups hidden behind the noise. Thus, using single view is unlikely to recover the inherent cluster structure. This phenomenon is also validated by the low NMI as well as ARI of these two views. Applying multiple views helps reduce the noise and uncover the shared cluster structure. Second, compared with the five other baseline multi-view clustering strategies, our tensor based clustering methods perform better. In particular, MC-FR-OI-MLSVD, MC-FR-MI-HOOI and MC-TR-I-EVDit are obviously superior to others based on both NMI and ARI evaluations. LMF performs wrongly on this data, and thus we omit its comparison.

To evaluate whether the optimized weights assigned to single-view data are correlated with their clustering performance, we compare the ranking of weighting coefficients obtained by MC-FR-MI-HOOI with the ranking of the corresponding clustering performance in Table II, where we list these weighting factors as $\alpha_i$ and we also list weighting factors by MC-TR-I-EVDit as $\beta_i$. The ranking of these optimal weights is generally consistent with the ranking of clustering performance. As shown, the top two largest coefficients correctly indicate the top two best single-view data ($A_2$ and $A_3$). Although the ranking of the top 2 weighting coefficients is not exactly consistent with the ranking of the corresponding performance, their coefficients are quite near (0.5288 in A2 and 0.5643 in A3).

D. Application on scientific documents analysis

In this Section, we apply our algorithms to the scientific analysis of the Web of Science (WoS) journal set. Our objective is to map these journals into different subjects using clustering algorithms.

1) Data description: Historically, bibliometric researchers have focused solely on citation analysis or text analysis, but not on both simultaneously. Recently, many researchers have applied text mining and citation analysis to the journal set analysis. The integration of lexical and citation information is a promising strategy towards better mappings [25]. We adopt a data set obtained from the WoS database by Thomson Scientific which contains articles, letters, notes and reviews from the years 2002 till 2006. To create a balanced benchmark data for evaluation, we select seven categories consisting of 1424 journals. The titles, abstracts and keywords of the journal publications are indexed by a Jakarta Lucene based text mining program using no controlled vocabulary. The weights are calculated by four weighting schemes: TF-IDF, IDF, TF and binary. Therefore, we have obtained four data sources as the lexical information of journals. These four kinds of text data are directly represented as similarity matrices. At the same time, four kinds of citation data represent link-based relationships among journals and consequently, from them, we construct corresponding affinity matrices, denoted as cross-citation, co-citation, bibliographic coupling and binary cross-citation. The details of journal data are presented in Supplementary material 1.

We implement the proposed tensor based multi-view clustering methods to integrate multi-view data on journal data. To evaluate the performance, we also apply six popular multi-view clustering methods mentioned in Section V to integrate multi-view data. To verify whether the integration of multi-view data by tensor methods indeed improves the clustering performance, we first systematically compare the performance of all the individual data sources using spectral clustering. As shown in the left part of Table III, the best spectral clustering is obtained on TFIDF data (NMI 0.7280, ARI 0.6601). Next, we implement our tensor based multi-view clustering on different types of multi-view data integrations detailed in Supplementary material 2. Text data and citation data are heterogeneous data because they are generated from various feature spaces (see clustering results of their integration from Table II to Table IV). Multi-view data solely from text or citation is homogeneous because it shares the same feature space (see...
clustering results of homogeneous integration of both text data from Table V to Table VII and citation data from Table VIII to Table X. As shown, the best multi-view clustering performance is obtained from MC-FR-MI-HOOI by integrating two homogeneous text data of TFIDF and IDF (NMI 0.8201, ARI 0.8229). Moreover, we also find that the clustering performance of different integration schemes varies significantly based on the choice of single-view data. This implies that to some degree, the multi-view clustering performance depends on the quality of the single-view data involved. For instance, in the best multi-view clustering case above, TFIDF and IDF are the two single-view data sources with the two best clustering performance.

Afterwards, we also investigate the performance of integrating all single-view data using all compared multi-view clustering presented in the right part of Table III. In particular, of all the methods we compared, the best performance is obtained by the MC-FR-OI-HOOI method (NMI 0.7605, ARI 0.7262).

The comparison between the ranking of weighting coefficients by MC-MI-HOOI with the ranking of clustering performance is shown in Table IV, where we list these weighting factors as $\alpha_i$ and we also list weighting factors by MC-TR-I-EVDit as $\beta_i$. Because text and citation data are heterogeneous data sources, we separately compare each integration of each type of data in its own feature space. In general, the ranking of these optimal weights is consistent with the ranking of their individual performance. For instance, within the citation feature space, the top two largest coefficients correctly indicate the top two best individual data source (co-citation and cross-citation). In addition, we can see although the values of these weighting factors by MC-FR-MI-HOOI are different from the counterparts by MC-TR-I-EVDit, the ranking of weighting factors by MC-FR-MI-HOOI is the same to that by MC-TR-I-EVDit.

In Figure 9, two confusion matrices of journal data are depicted to illustrate the partition difference between our multi-view clustering result (by MC-FR-OI-HOOI) and the best single-view clustering result (on TFIDF data). The values of the matrices are normalized according to $R_{ij} = C_{ij}/T_i$, where $T_j$ is the total number of journals belonging to standard label of ESI category $i$ and $C_{ij}$ is the number of these $T_j$ journals that are clustered to class $j$. The results show that the intuitive confusion matrices correspond to the numerical evaluation results. For instance, the quality of clustering obtained by MC-FR-OI-HOOI (NMI 0.7605, ARI 0.7262) is higher than that of spectral clustering on TFIDF. In the confusion matrix of spectral clustering on TFIDF, 15 journals belonging to Agriculture Science (Nr. 1) are mis-clustered to Environment Ecology (Nr. 3), and 60 journals are mis-clustered to Pharmacology and toxicology (Nr. 7). Meanwhile, by MC-FR-OI-HOOI, the number of Agriculture Science (Nr. 1) journals mis-clustered to Environment Ecology is reduced to 7, and the number to Pharmacology and Toxicology is reduced to 26.

E. Experiment on disease gene clustering

Text mining helps biologists automatically collect disease-gene associations from large volumes of biological literature. Given a list of genes, we can generate a gene-by-term matrix by the retrieval from the medical literature analysis and retrieval system online (MEDLINE) database. We can also obtain multi-view gene-by-term matrices. The view represents a text mining result retrieved by specific controlled vocabularies, hence multi-view text mining is featured as applying multiple controlled vocabularies to retrieve the gene-centric perspectives from free text publications. The clustering methods can be implemented on these genes to get the group information, which can be utilized for further disease analysis.

The data sets contain ten different gene-by-term text profiles indexed by ten controlled vocabularies. The original disease-related gene data set contains 620 genes that are known to be relevant to 29 diseases. To avoid the effect of imbalanced clusters that may affect the evaluation, we only keep the diseases that have 11 to 40 relevant genes. This step results in 14 genetic diseases and 278 genes. Because the present paper focuses on non-overlapping ("hard") clustering, we additionally remove 16 genes that are relevant to multiple diseases and 17 genes whose term vectors are empty for one of these ten vocabularies. The remaining 245 disease relevant genes are clustered into 14 clusters and biologically evaluated by their disease labels. For each vocabulary based gene-by-term data source, we create a similarity matrix using the value of the cosine similarity for two vectors. The details of the disease gene data analysis are presented in Supplementary material 3.

At first, as shown in the left part of Table V, the best clustering performance of individual data sources is obtained on LDDB text mining profile (NMI 0.7088, ARI 0.5942). Next, we also implement 45 types of integration of multi-view text mining data for clustering. The clustering performance is presented in Supplementary material 4 from Table XIII to Table XV. As shown, the best clustering performance is obtained by MC-FR-OI-HOOI through integrating multi-view data by GO, MeSH, OMIM, NCI, eVOC, KO, LDDB and MP (NMI 0.7687, ARI 0.6364). Afterwards, we also investigate the clustering performance of integrating all single-view data using all the multi-view clustering methods presented in the right part of Table V. In particular, among all the relevant clustering methods, the best performance is still obtained by the MC-FR-OI-HOOI method (NMI 0.7732, ARI 0.6473) as analyzed in the former experiment on journal data. The multi-view clustering strategies with the next two best performance are still our tensor methods, MC-FR-MI-HOOI (N-MI 0.7494, ARI 0.6015) and MC-FR-OI-MLSVD (NMI 0.7429,
βMC-TR-I-EVDit as α weighting factors as their corresponding clustering performance, where we list these to the six baseline multi-view clustering methods, demonstrating spectral clustering results of any single-view data but also superior ARI 0.6030). All of our tensor based methods are not only beyond the power of our multi-view clustering strategy. In Table VI, we present the comparison between the ranking of weighting coefficients among multi-view data with the ranking of their corresponding clustering performance, where we list these weighting factors as αi and we also list weighting factors by MC-TR-I-EVDit as βj. As shown, the largest coefficient correctly indicates the best individual data source (LDDB), while the smallest coefficient correctly indicates the worst individual data source (KO). As a whole, the ranking of these optimal weights are consistent with the ranking of the corresponding performance.

In addition, we can see although the values of these weighting factors by MC-MI-HOOI are different from the counterparts by MC-TR-I-EVDit, the ranking of weighting factors by MC-FR-MI-HOOI is almost the same to that by MC-TR-I-EVDit. In Figure 10, two confusion matrices of disease gene data are depicted to illustrate the partition difference between our multi-view clustering (by MC-FR-OI-HOOI) and the best single-view clustering (by MC-TR-I-EVDit), the ranking of weighting factors by MC-FR-OI-HOOI are different from the counterparts by MC-MI-HOOI is almost the same to that by MC-TR-I-EVDit.

**Q1.1** In spectral clustering, checking the "elbow" of the plot of the eigenvalues of single-view data provides a heuristic evaluation results. As shown in Table V, the quality of clustering obtained by MC-FR-OI-HOOI (NMI 0.7605, ARI 0.7262) is the best individual data source (LDDB), while the middle and right parts of Figure 11 show the elbow plots for the journal data and the disease data, respectively. For our analysis we used the cluster numbers indicated by the arrows. Moreover, in Figure 1 and Figure 2 of Supplementary material 5, we also compare the estimate of the number of clusters [28]. Analogous, in our tensor approach the plot of mode-1 singular values of the similarity tensor provides a heuristic estimate of the number of clusters. In Figure 11 we plot the 20 dominant mode-1 singular values of the similarity tensor for our three data sets. The elbow for the synthetic data is between 2 and 4. The real number of clusters is 3. The middle and right parts of Figure 11 show the elbow plots for the journal data and the disease data, respectively. For our analysis we used the cluster numbers indicated by the arrows. Moreover, in Figure 1 and Figure 2 of Supplementary material 5, we also compare the
VI. DISCUSSION

Based on the clustering performance of the multi-view clustering strategies, first, MKF is efficient when compared with tensor based strategies. However, MKF only combines multiple kernels (similarity matrices) in a simple way using the average sum of multiple similarities. Thus, such a simple combination neglects the discriminating capability of each kernel. Second, clustering ensemble methods (SA and AdacVote) rely on discrete hard clustering. Using only the final partition information seems too fragile to integrate.

In addition, because the partition of every single-view data is required, the implementation of clustering ensemble methods is not efficient as shown in Table VII.

Third, considering LMF, we found that the clustering performance relies on the initialization, and hence the partition results are quite unstable. Moreover, its optimization mechanism consumes much time.

Fourth, for CP-ALS, the failure might be due to the un-orthogonal property of the relaxed assignment matrix \( U \) after tensor decomposition. The reason is that the similarity matrix in (3) we adopted to construct the tensor corresponds to the Ncut based Laplacian matrix that requires the orthogonal partition in spectral clustering.

Meanwhile, our tensor based multi-view spectral clustering can be thought of as a “Multi-view PCA” analysis, which integrates multi-view information seamlessly and forms a joint optimal subspace. Therefore our strategy can extract the latent pattern without the discri minating capability of each kernel. Consequently, with number of views increasing, the computation of the clustering ensemble method will become more and more intensive.
VII. CONCLUSION AND OUTLOOK

We proposed a multi-view clustering framework based on high-order analogues of the matrix Singular Value Decomposition (SVD) and Principal Component Analysis (PCA). Our framework can be regarded as a multi-view extension of spectral clustering. With our tensor formulation, both heterogeneous and homogeneous information can be integrated to facilitate the clustering task.

We presented two new multi-view clustering strategies: multi-view clustering by the integration of the Frobenius-norm objective function (MC-FR-OI) as well as the matrix integration in the Frobenius-norm objective function (MC-FR-MI). The relevant tensor based solutions are proposed, which are either iterative optimization or efficient approximation. All of them are capable of utilizing the global information of multi-view data while taking the effect of single-view data into consideration. Furthermore, these different methods can be applied to various practical scenarios.

We employed our algorithms to both synthetic data and two real applications. The clustering performance demonstrated that our algorithms are not only superior to single-view spectral clustering methods, but also superior to other baseline multi-view clustering methods.

In later research, we will carry out our work in the following directions: (1) We will investigate other alternative tensor solutions, such as INDSCAL [4], as well as efficient tensor decomposition for scalable application; (2) We will extend our multi-view clustering algorithm to higher-order data (we only use three-order data in this research), such as, adding another temporal order that allows data to vary at different time points; (3) Our framework is not limited to the clustering analysis. Since its core is to seek a joint optimal latent subspace, it can be extended to other multi-view learning tasks: for instance, classification, spectral embedding, collaborative filtering and even information retrieval.

ACKNOWLEDGMENT

The authors would like to thank L. De Lathauwer for deriving the version of HOOI with a single vector in one mode and for the theorem and proof in the Supplementary material 6. This work was supported by (1) National Natural Science Foundation of China (Grant No. 61105058); (2) The joint post-doctoral programme by Credit Reference Center and Financial Research Institute, The People’s Bank of China; (3) China Postdoctoral Science Foundation funded project; (4) The related fundings: GOA-AMBioRICS, CoE-EF/05/006(OPTEC), IOF-SCORES4CHEM, FWO-G0452.04, FWO-G0499.04, FWO-G0211.05, FWO-G0226.06, FWO-G0321.06, FWO-G0302.07, FWO-ICCoS, FWO-ANMMM, FWO-MLDLM, IWT-McKown-E, Eureka-Fliteplus, IAP-P6/04, ERNSI; (5) Flemish Government: Center for R&D Monitoring.

REFERENCES
