Multi-View Partitioning via Tensor Methods

Xinhai Liu, Lieven De Lathauwer, *Senior Member, IEEE*, 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 optimization integration (MC-OI) and that based on matrix integration (MC-MI). 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, heterogeneous data integration, tensor decomposition, spectral clustering, multi-linear singular value decomposition, higher-order orthogonal iteration

I. INTRODUCTION

In many real-world problems, objects 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 [18]. In multiplex network analysis, we are given a set of multiple networks that share the same nodes but possess network-specific links representing different types of relationships between nodes [21]. 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 view of the same set of underlying objects. A simple approach to learn from these multi-view data is to learning from each view separately. However, such approaches fail to account for the complementary information encoded into different views.

Multi-view clustering refers to the clustering of the same class of entities with multi-view representations, either from various information sources or from different feature generators. Compared with the clustering that is implemented on single-view data, multi-view clustering is

expected to obtain 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 few years [33]. 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 [20] works well on single-view data, it is not well suited for the presentation of multi-view data, since it is inherently based on matrix decompositions.

Recently, several multi-view clustering algorithms have been proposed [1], [3], [5], [18], [19], [27], [30], [35]. These multi-view clustering techniques have been shown to yield better performance in comparison to single-view techniques. However, as we will discuss in the related work, the limitations of some algorithms are apparent. 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 [27], [31]. A unified framework that can integrate various types of multi-view data is lacking [19], [30].

Traditionally, tensor-based methods have been used to model multi-view data [15]. Tensors are higher-order generalizations of matrices, and some tensor methods are very powerful to analyze the latent pattern hidden in the 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 naturally to extract hidden (latent) components (cluster structure) and investigate complex relationship among them. Tensors have been successfully applied to several domains, such as chemometrics, signal processing, Web search, data mining and scientific computing [9], [15], [16], [25], [28].

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 (MC-OI) and that based on matrix integration (MC-MI). In particular, MC-MI can assign each view a suitable weight to boost the clustering. For each strategy, we provide the relevant tensor based solutions. In fact, just as the other variants of PCA in machine learning applications [34], our strategy can be taken as a multi-view PCA analysis.

As an illustrative example of synthetic data shown in Figure 1, we intend to use it to compare three partition strategies to show the power of our tensor based multi-view partition. There are two groups of data points in a 3-D space, suppose based on our limited measurements (such as 2-D cameras in the real world), only the 2-D projection information of these data points could be observed (such as, X-Y projection, Y-Z projection and X-Z projection in a 3-D X-Y-Z coordinate system). We call each of the three 2-D projection data, single-view data. We can find the group information by adopting a spectral partition on each of these three single-view data. As shown in the upper right part of Figure 1, from each spectral projection of single-view data, there is significant overlap among these two groups of data points. Obviously, we could not recover the cluster structure only by each single-view data.

On the other hand, a natural idea is to integrate these three-view observations for joint partitioning, that is multi-view clustering. Thus, we investigate the spectral projection of two multi-view clustering strategies: multiple kernel fusion (MKF) and MC-OI based on MLSVD (MC-OI-MLSVD). As shown in the middle right part of Figure 1, the overlap among these two groups still remains by the spectral projection of MKF so that it is hard to get the correct group structure as well. Whereas, from the lower right part of Figure 1, the two groups are separated clearly by MC-OI-MLSVD and consequently a good group structure can be recovered from this three-view data. In this example, it shows that our tensor based multi-view clustering strategy is able to obtain the latent cluster structure hidden amid multi-view data.

To the best of our knowledge, our work is the first unified attempt to address multi-view

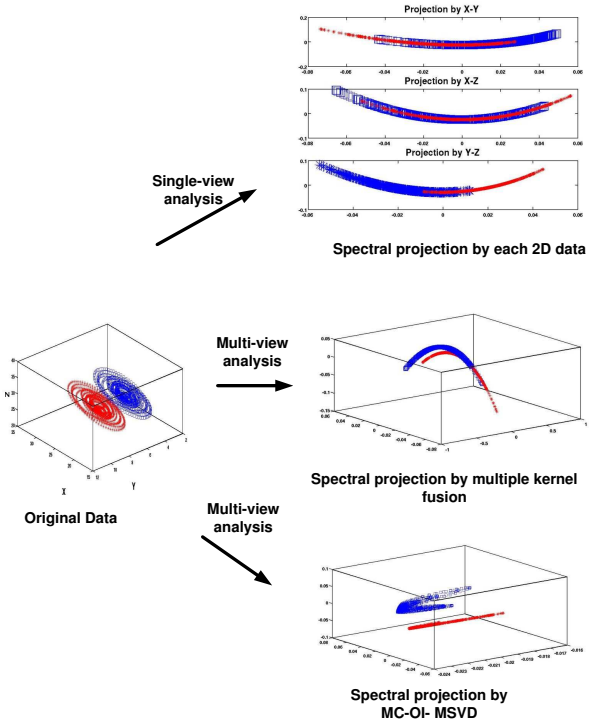

Fig. 1. Comparison of single-view projection versus multi-view projection.

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.

Notations: 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 $(\mathcal{A}, \mathcal{B}, \ldots)$.

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.

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*. [19] 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 [31]. Zhou *et al*. [35] 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.

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 [30]. Based on modularity optimization, Mucha *et al.* [21] 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 [14], [18]. Such a solution might sound natural, but its underlying principal is

not clear [35]. In addition, the heavy computation of their convex optimization makes them only applicable to small databases [18]. Meanwhile, with the recent research progress in tensor decomposition [24], 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], [27]. 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* [28] 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.* [9] 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 [25]. 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 [26]. In particular, spectral clustering involves a matrix trace optimization problem [20], [22]. We show in this paper that the spectral clustering formalism can be extended to deal with multi-view problems based on tensor computations.

Given a set of N data points $\{x_i\}_{i=1}^N$ where $x_i \in \mathbb{R}^d$ is the ith data point, a similarity $s_{ij} \geq 0$ can be defined for each pair of data points x_i and x_j based on some similarity measure. An intuitive way to represent this data set is using a graph $G = (V, E)$ in which the vertices V represent the data points and the edges $e_{ij} \in \varepsilon$ 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 $\mathbf{S}$ with the ijth entry $\mathbf{S}_{ij} = s_{ij}$. The degree of the vertex v_i, defined as

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

is the sum of all the weights of edges connected to v_i. The degree matrix $\mathbf{D}$ is a diagonal matrix containing the vertex degrees $d_1, ..., d_N$ on the diagonal. It follows from the spectral embedding formalism [20], [22], [26] that the Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{S}$, and the normalized Laplacian matrix, corresponding to the normalized cuts (Ncut), is defined as

$$\mathbf{L}_{\text{Ncut}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{S}_N, \tag{2}$$

where $\mathbf{S}_N$ is the normalized similarity matrix and defined as

$$\mathbf{S}_N = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}. \tag{3}$$

The matrices $\mathbf{S}_N$ and $\mathbf{L}_{\text{Ncut}}$ have the same eigenvectors, and their eigenvalues are related as $\lambda^{(\mathbf{S}_N)} = 1 - \lambda^{(\mathbf{L}_{\text{Ncut}})}$, where $\lambda^{(\mathbf{S}_N)}$ and $\lambda^{(\mathbf{L}_{\text{Ncut}})}$ are the eigenvalues for $\mathbf{S}_N$ and $\mathbf{L}_{\text{Ncut}}$, respectively.

A. Single-view spectral clustering

We first consider spectral clustering in the single-view setting. Suppose $\mathbf{U} \in \mathbb{R}^{N\times M}$ is the relaxed assignment matrix, where N is the number of data points and M is the number of clusters. The spectral clustering problem can be expressed as

$$\begin{aligned} &\min_{\mathbf{U}} \ \text{tr}(\mathbf{U}^T\mathbf{L}_{\text{Ncut}}\mathbf{U}), \\ &\text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{4}$$

It follows from the Ky Fan theorem [23] that the optimal solution to the problem in (4) is the top M eigenvectors of $\mathbf{L}_{\text{Ncut}}$. Considering the relationship between $\mathbf{S}_N$ and $\mathbf{L}_{\text{Ncut}}$, the spectral clustering formulation can also be expressed as

$$\begin{aligned} &\max_{\mathbf{U}} \ \text{tr}(\mathbf{U}^T\mathbf{S}_N\mathbf{U}), \\ &\text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{5}$$

Since $\mathbf{S}_N$ is positive semi-definite, the spectral clustering can be re-formulated as a Frobenius norm optimization problem as follows:

$$\begin{aligned} &\max_{\mathbf{U}} \ \|\mathbf{U}^T\mathbf{S}_N\mathbf{U}\|_F^2, \\ &\text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{6}$$

The objective functions in (5) and (6) are different, but they happen to have the same optimal solution, namely, the columns of a matrix $\mathbf{U}$ span the dominant eigenspace of $\mathbf{S}_N$.

B. Multi-view spectral clustering

Given multi-view data, the clustering result could be improved if the multiple views are integrated in an appropriate way. We have the following two strategies to integrate the multi-view data in the context of spectral clustering. Our multi-view partition strategies are expected to capture the complementary information conveyed in different views so that they are able to achieve better or robust clustering results.

1) Multi-view clustering by optimization integration (MC-OI): Based on the spectral partition of each single-view data, the first strategy is to integrate the objective functions of individual partitions from each single-view data. In particular, we consider the optimization of multi-view clustering by simply adding individual objective functions as in

$$\begin{aligned} \max_{\mathbf{U}} \ & \sum_{i=1}^{K} \|\mathbf{U}^T \mathbf{S}_N^{(i)} \mathbf{U}\|_F^2, \\ & \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}, \end{aligned} \tag{7}$$

where $\mathbf{S}_N^{(i)}$ is the normalized similarity matrix for the ith view and $\mathbf{U}$ is the common factor shared by multiple views.

2) Multi-view clustering by matrix integration (MC-MI): The second multi-view clustering strategy is to combine the normalized similarity matrices from different views, leading to the following integrated similarity matrix as

$$\tilde{\mathbf{S}} = \alpha_1 \mathbf{S}_N^{(1)} + \alpha_2 \mathbf{S}_N^{(2)} + ... + \alpha_K \mathbf{S}_N^{(K)}, \tag{8}$$

where α_i are the weights of each view and $W = \{\alpha_1, \alpha_2, \cdots, \alpha_K\}^T$. The multi-view clustering based on $\tilde{\mathbf{S}}$ can be formulated as follows:

$$\begin{aligned} & \max_{\mathbf{U}, \alpha_i} \ \|\mathbf{U}^T \tilde{\mathbf{S}} \mathbf{U}\|_F^2, \\ \text{s.t. } & \mathbf{U}^T\mathbf{U} = \mathbf{I}, \ \alpha_i > 0 \text{ and } \sum_{i=1}^{K} \alpha_i^2 = 1, \end{aligned} \tag{9}$$

where the unknown weighting factors α_i play a crucial role in the above optimization. Once α_i are determined, MC-MI can be handled as a common spectral clustering problem defined in

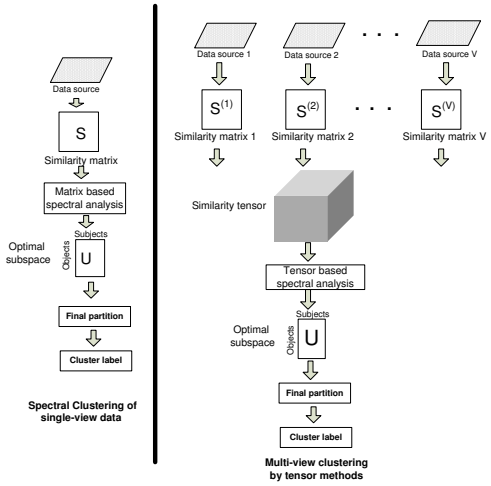

Fig. 2. Comparison between single view (left) and multi-view (right) spectral clustering.

(6). In addition, weighting factors α_i can also be considered as the weights (contribution) of different views during joint partition.

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

We provide some basic background on tensors and their decompositions in the following. We refer the readers to [6], [7], [16] for more detailed treatment on this topic. A tensor is a multi-way array [16]. 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 investigate third-order tensor methods that are relevant to our problem.

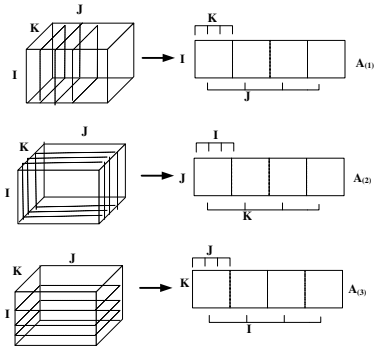

Fig. 3. Matrix unfolding of a third-order tensor

Matrix unfolding is the process of re-ordering the elements of a 3-way array into a matrix. The n-mode $(n = 1, 2, 3)$ matrix unfoldings of a tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ are denoted by $\mathbf{A}_{(1)}, \mathbf{A}_{(3)}$ and $\mathbf{A}_{(3)}$ separately. For example, the matrix unfolding $\mathbf{A}_{(1)}$ is a matrix with the number of rows I and the number of its columns is the product of dimensionalities of all other modes, that is, $J \times K$. The matrix unfolding of a third-order tensor is illustrated in Figure 3.

A tensor can be multiplied by a matrix. Consider a tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ and a matrix $\mathbf{B} \in \mathbb{R}^{I_1 \times I}$, $\mathbf{C} \in \mathbb{R}^{J_1 \times J}$, $\mathbf{D} \in \mathbb{R}^{K_1 \times K}$, then the 1-mode product $(\mathcal{A} \times_1 \mathbf{B})$, 2-mode product $(\mathcal{A} \times_2 \mathbf{C})$ and 3-mode product $(\mathcal{A} \times_3 \mathbf{D})$ are defined by

$$(\mathcal{A} \times_1 \mathbf{B})_{i_1 jk} = \sum_{i_1=1}^{I_1} a_{ijk} b_{i_1 i}, \ \forall i_1, j, k, \tag{10}$$

$$(\mathcal{A} \times_2 \mathbf{C})_{ij_1 k} = \sum_{j_1=1}^{J_1} a_{ijk} c_{j_1 j}, \ \forall i, j_1, k, \tag{11}$$

$$(\mathcal{A} \times_3 \mathbf{D})_{ijk_1} = \sum_{k_1=1}^{K_1} a_{ijk} d_{k_1 k}, \ \forall i, j, k_1, \tag{12}$$

respectively.

Multi-linear singular value decomposition (MLSVD) [6], [32] is a form of higher-order extension of matrix singular value decomposition (SVD). It decomposes a tensor into a core tensor multiplied by a matrix along each mode. In the three-way case where $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$, we have

$$\mathcal{A} = \mathcal{B} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}, \tag{13}$$

where $\mathbf{U} \in \mathbb{R}^{I \times I}$, $\mathbf{V} \in \mathbb{R}^{J \times J}$ and $\mathbf{W} \in \mathbb{R}^{K \times K}$ are called factor matrices or factors and can be thought of as the principal components of the original tensor along each mode. The factor matrices $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$ are assumed to be column-wise orthonormal. The tensor $\mathcal{B} \in \mathbb{R}^{I \times J \times K}$ is called the core tensor. In MLSVD, $\mathcal{B}$ has a very specific structure, namely, it satisfies "all-orthogonal" and "ordering" constraints, see [6]. The elements of $\mathcal{B}$ show the level of interaction between different components. According to [6], given a tensor $\mathcal{A}$, its matrix factors $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$ as defined in (13) can be computed as the left singular vectors of its matrix unfoldings $\mathbf{A}_{(1)}, \mathbf{A}_{(2)}$ and $\mathbf{A}_{(3)}$ respectively. The tensor approximation by truncating the decomposition is named truncated MLSVD. A decomposition as (13), with or without constraint, is also known as a Tucker decomposition [32].

B. Tensor construction

As aforementioned in the introduction Section, multi-view data can be naturally modelled as a tensor. The construction of a tensor is a key step to devise our multi-view clustering algorithm. There are several options for constructing a tensor with multi-view data. In [11], 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 our later application to scientific publication analysis, the citation feature space has the dimension of 8,305 while the dimension of the text feature space is more than 600,000.

Consequently, we prefer a construction that is independent of data dimension, thereby enabling the integration of heterogeneous data sources. Based on this motivation, we propose to build a tensor $\mathcal{A}$ from the multiple similarity matrices $\{\mathbf{S}_N^{(1)}, \mathbf{S}_N^{(2)}, \cdots, \mathbf{S}_N^{(K)}\}$ derived from multiple views as the frontal slices. The construction of a similarity tensor is illustrated in the right part of Figure 4. Since the similarity of each view is computed in different spaces, the normalization of each similarity matrix is required. Indeed, our definition of similarity matrix in (3) could be regarded as a normalization step.

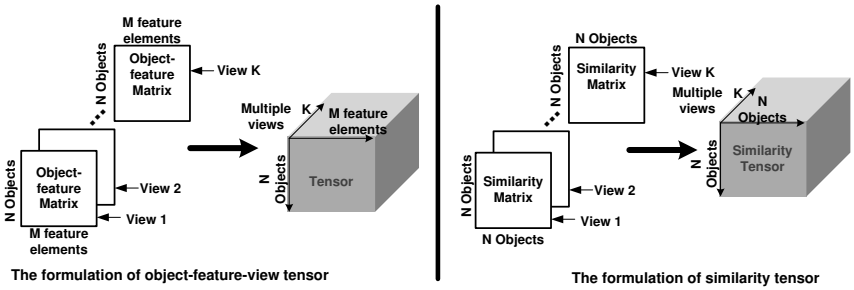

Fig. 4. Comparison of different formulations of multi-view learning using tensor methods.

C. MC-OI by tensor methods

We first discuss the optimization integration approach for multi-view clustering. Suppose a similarity tensor $\mathcal{A}$ is built from similarity matrices $\mathbf{S}_N^{(i)} \in \mathbb{R}^{N\times N}$ $(i = 1, \ldots, K)$, the integration of spectral optimization can be written as

$$\sum_{i=1}^{K} \|\mathbf{U}^T \mathbf{S}_N^{(i)} \mathbf{U}\|_F^2 = \|\mathcal{A} \times_1 \mathbf{U}^T \times_2 \mathbf{U}^T \times_3 \mathbf{I}\|_F^2, \tag{14}$$

where the column space of $\mathbf{U} \in \mathbb{R}^{N\times M}$ is the joint optimal subspace and $\mathbf{I} \in \mathbb{R}^{K\times K}$ is an identity matrix. The spectral decomposition of the similarity tensor $\mathcal{A}$ in this case can be illustrated as in Figure 5.

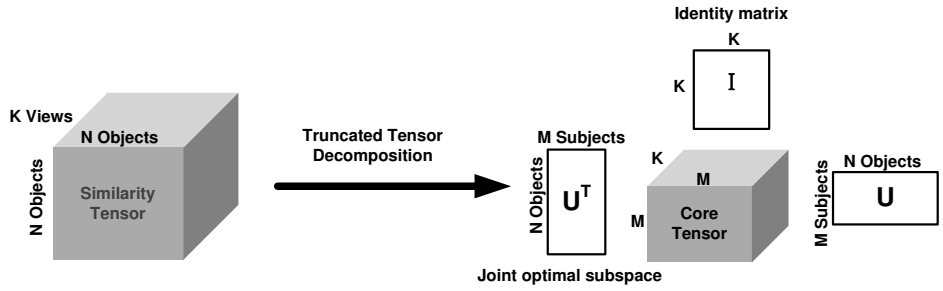

Fig. 5. Illustration of multi-view clustering by optimization integration using tensor decomposition.

The optimization of multi-view clustering in (7) can be re-formulated based on tensor com-

putation as

$$\max_{\mathbf{U}} \|\mathcal{A} \times_1 \mathbf{U}^T \times_2 \mathbf{U}^T \times_3 \mathbf{I}\|_F^2, \quad \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}. \tag{15}$$

This optimization can be approximated by MLSVD and we call this method multi-view clustering based on MLSVD (MC-OI-MLSVD). As explained in [6], projection by MLSVD on the dominant higher-order singular vectors usually gives a good approximation of the given tensor. Consequently, we propose to take the columns of $\mathbf{U}$ to be the dominant 1-mode singular vectors. Our experimental results show that this solution usually leads to satisfactory performance. The dominant 1-mode singular vectors of $\mathbf{U}$ equal to the dominant left singular vectors of $\mathbf{A}_{(1)}$. The truncated MLSVD obtained this way does not maximize (7) in general. However, the result is usually satisfactory and the algorithm is efficient and easy to implement. Moreover, there exists an upper bound on the approximation error as shown in [6]. The pseudo code of MC-OI-MLSVD is presented as follows:

Algorithm IV.1: MC-OI-MLSVD($\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, ..., \mathbf{S}^{(K)}, M$)

comment: M is the number of clusters

1. *Build a similarity tensor* $\mathcal{A}$
2. *Obtain the unfolding matrix* $\mathbf{A}_{(1)}$
3. *Compute* $\mathbf{U}$ *from the subspace spanned by the dominant left* M *singular vectors of* $\mathbf{A}_{(1)}$
4. *Normalize the rows of* $\mathbf{U}$ *to unit length*
5. *Calculate the cluster idx with* $k-means$ *on* $\mathbf{U}$

return ($idx:$ *the clustering label*)

Meanwhile, there exist other tensor based solutions. For example, this tensor based optimization can be solved by a tensor approximation method called higher-order orthogonal iteration (HOOI), which is an alternating least-squares (ALS) algorithm [7], [17]. In addition, other algorithms can also be employed to solve this tensor based optimization problem, see [13] and reference herein.

The basic idea of HOOI is to solve the following maximization problem:

$$\max_{\mathbf{U},\mathbf{V},\mathbf{W}} \|\mathcal{A} \times_1 \mathbf{U}^T \times_2 \mathbf{V}^T \times_3 \mathbf{W}^T\|_F^2, \quad \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I},\ \mathbf{V}^T\mathbf{V} = \mathbf{I} \text{ and } \mathbf{W}^T\mathbf{W} = \mathbf{I}. \tag{16}$$

At each step, the estimate of one of the matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$ is optimized, while the other two are fixed. In order to maximize with respect to the unknown matrix $\mathbf{U}$, the objective function of (16) is treated as a quadratic expression in $\mathbf{U}$. It follows from

$$\|\mathcal{A} \times_1 \mathbf{U}^T \times_2 \mathbf{V}^T \times_3 \mathbf{W}^T\|_F^2 = \|\mathbf{U}^T(\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W}))\|_F^2, \tag{17}$$

where the columns of $\mathbf{U} \in \mathbb{R}^{I \times P}$ build an orthonormal basis for the P-dimensional left dominant subspace of the column space of $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$, and the solution can be obtained from the SVD of $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$ [6]. The optimization with respect to $\mathbf{V}$ and $\mathbf{W}$ is performed in similar ways. Usually, the optimization of HOOI is initialized by MLSVD. Thus, the resulting algorithm called MC-OI-HOOI, is presented as follows:

Algorithm IV.2: MC-OI-HOOI($\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, ..., \mathbf{S}^{(K)}, M$)

Build a similarity tensor $\mathcal{A}$

Obtain the unfolding matrices $\mathbf{A}_{(1)}$, $\mathbf{A}_{(2)}$ *and* $\mathbf{A}_{(3)}$

comment: obtain an initial $\mathbf{U}_0$ and $\mathbf{V}_0$ by MLSVD

while $<!convergence>$

do
$$\begin{cases} step1 : \mathbf{U}_{i+1} \text{ in dominant subspace of} \\ \mathbf{A}_{(1)}(\mathbf{V}_i \otimes \mathbf{I}) \\ step2 : \mathbf{V}_{i+1} \text{ in dominant subspace of} \\ \mathbf{A}_{(2)}(\mathbf{U}_i \otimes \mathbf{I}) \end{cases}$$

comment: i is the counter of iteration

Normalize the rows of $\mathbf{U}$ *to unit length*

Calculate the cluster idx with $k-means$ *on* $\mathbf{U}$

return $(idx : \text{ the clustering label})$

The tensor decomposition in both MC-OI-MLSVD and MC-OI-HOOI is in fact a kind of joint matrix compression as shown in Figure 6, where the truncated tensor decomposition in the upper part can be understood as the joint compression of a set of matrices in the lower part. Matrix $\mathbf{U}$ in both parts is the common factor shared among multi-view data, and a set of similarity matrices $\Lambda^{(1)}, \ldots, \Lambda^{(K)}$, which form the front slices of the core tensor in the upper part, capture the characteristics of each view.

D. MC-MI by tensor methods

Since the effect of multi-view data differs from each other, each of them can be assigned an appropriate weight for joint analysis. In this case, the spectral decomposition of the similarity

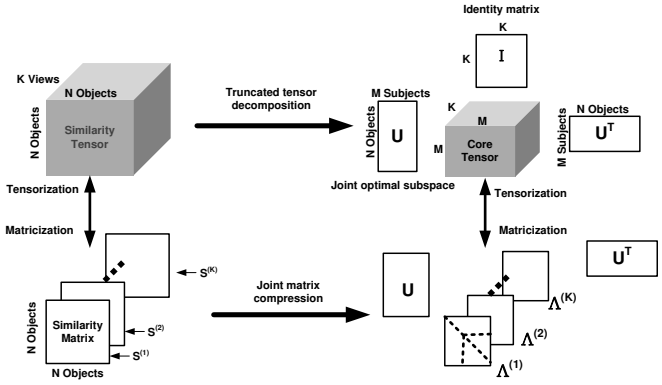

Fig. 6. Illustration of the joint matrix decomposition for multi-view data.

tensor $\mathcal{A}$ can be illustrated in Figure 7, in which $\mathbf{U}$ denotes the joint optimal subspace that we want to compute, and $W = \{\alpha_1, \alpha_2, \cdots, \alpha_K\}^T$ denotes the weights of each view. The objective function of multi-view clustering of matrix integration (MC-MI) can be written as

$$\|\mathbf{U}^T\tilde{\mathbf{S}}\mathbf{U}\|_F^2 = \|\mathbf{U}^T(\sum_{i=1}^{K}\alpha_i\mathbf{S}_N^{(i)})\mathbf{U}\|_F^2 = \|\mathcal{A}\times_1\mathbf{U}^T\times_2\mathbf{U}^T\times_3 W^T\|_F^2. \tag{18}$$

Thus the optimization of multi-view clustering in (9) can be re-written as

$$\max_{\mathbf{U},W}\ \|\mathcal{A}\times_1\mathbf{U}^T\times_2\mathbf{U}^T\times_3 W^T\|_F^2,\ W = \begin{pmatrix}\alpha_1\\ \vdots \\ \alpha_K\end{pmatrix} \quad \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I},\ \|W\|_F^2 = 1. \tag{19}$$

We can obtain the solution of this weighted multi-view clustering by HOOI and we call this method MC-MI-HOOI.

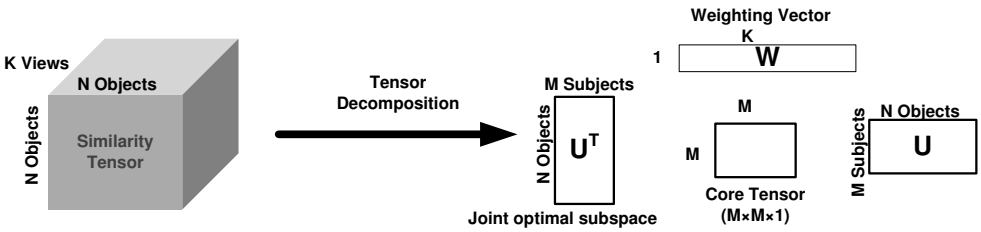

Fig. 7. Illustration of multi-view clustering by matrix integration using tensor decomposition

Since the optimal HOOI solution usually leads to expensive computation, we simplify the main steps of the HOOI solution by an equivalent but efficient implementation. Taking the fact that *W* is not a matrix but a vector into account, we replace the optimization of the decomposed factors in each mode with the sum of matrices and its eigenvalue decomposition (EVD). The pseudo code of MC-MI-HOOI is shown in the following:

Algorithm IV.3: MC-MI-HOOI($\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, ..., \mathbf{S}^{(K)}, M$)

Build a similarity tensor $\mathcal{A}$

Obtain the unfolding matrices $\mathbf{A}_{(1)}$, $\mathbf{A}_{(2)}$ *and* $\mathbf{A}_{(3)}$

comment: obtain an initial $\mathbf{U}_0$ by MLSVD

while $<!convergence>$

do
- $step1 : Calculating\ W_i\ as\ the\ dominant\ left\ singular\ vector\ of\ \mathbf{A}_{(3)}(\mathbf{U}_i \otimes \mathbf{U}_i)$
- $step2 : Computing\ a\ new\ integration\ matrix\ \tilde{\mathbf{S}}\ by\ \sum_i^K W_i \mathbf{S}^{(i)}$
- $step3 : Obtaining\ \mathbf{U}_{i+1}\ by\ eigenvalue\ decomposition\ of\ \tilde{\mathbf{S}}$

comment: i is the counter of iteration

Normalize the rows of $\mathbf{U}$ *to unit length*

Calculate the cluster idx with $k-means$ *on* $\mathbf{U}$

return $(idx :\ the\ clustering\ label)$

In the MC-OI framework, we discussed two alternative approaches, MC-OI-MLSVD and MC-OI-HOOI. In the present Section, we only discussed MC-MI-HOOI. The reason is that tests indicated that in the MC-MI framework mere truncation of the MLSVD, retaining only one vector in the third mode, often yields unsatisfactory results.

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

Below are the six baseline multi-view clustering methods employed.

- Multiple kernel fusion (MKF): Joachims *et al.* [14] integrate different kernels by linear combination for hybrid clustering. The similarity matrix defined in (3) can be regarded as a linear kernel as well.

- Feature integration (FI) [30]: 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) [27]: 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, hyper-graph partition and meta-clustering) based on graph partitioning are employed to obtain the combined partition.
- AdacVote [1]: Ayad & Kamel propose a cumulative vote weighting method (AdacVote) to compute an empirical probability distribution summarizing the clustering ensemble.
- CP-ALS [4], [10]: 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 [31], 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 [8]. 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-OI-HOOI and MC-MI-HOOI by MLSVD.

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) [12] and Normalized Mutual Information (NMI) [27]. 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 communties (clusters), which have 50, 100 and 200 members respectively [29]. 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 [21], 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.

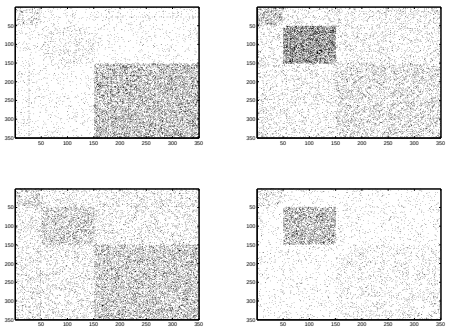

Fig. 8. Visualization of the adjacent matrices of a synthetic multiplex network.

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.

.

	Methods	NMI	ARI
Single View	S-A1	0.7605	0.7995
	S-A2	0.8928	0.9192
	S-A3	0.7198	0.8196
	S-A4	0.6318	0.5599
Multi View	MC-OI-MLSVD	*0.9321*	0.9508
	MC-OI-HOOI	0.9241	0.9509
	MC-MI-HOOI	**0.9431**	**0.9670**
	MKF	0.9156	0.9429
	FI	0.8893	0.9243
	SA	0.9251	*0.9540*
	AdacVote	0.8951	0.9400
	CP-ALS	0.5491	0.1274

TABLE I

EVALUATION OF CLUSTERING METHODS ON A FOUR VIEW SYNTHETIC MULTIPLEX NETWORK.

Sources	Ranking of α_i	α_i	Performance ranking
A1	3	0.4725	3
A2	2	0.5288	1
A3	1	0.5643	2
A4	4	0.4433	4

TABLE II

THE WEIGHTING COEFFICIENTS OF MULTI-VIEW DATA BY MC-MI-HOOI IN SYNTHETIC DATA.

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-OI-MLSVD and MC-MI-HOOI 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-

MI-HOOI with the ranking of the corresponding clustering performance in Table II. 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 [18]. 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 2 to Table 4). 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 5 to Table 7 and citation data from Table 8 to Table 10). As shown, the best multi-view clustering performance is obtained from MC-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-OI-HOOI method (NMI 0.7605, ARI 0.7262).

The comparison between the ranking of weighting coefficients by WC-MI-HOOI with the ranking of clustering performance is shown in Table IV. 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 Figure 9, two confusion matrices of journal data are depicted to illustrate the partition difference between our multi-view clustering result (by MC-OI-HOOI) and the best single-view clustering result (on TFIDF data). The values of the matrices are normalized according to $R_{ij} = C_j/T_i$, where T_i is the total number of journals belonging to standard label of ESI category i and C_j is the number of these T_i journals that are clustered to class j. The results show that

SC-Algorithm	NMI	ARI	MC-Algorithm	NMI	ARI
S-TFIDF	**0.7280**	**0.6601**	MC-OI-MLSVD	0.7331	0.6615
S-IDF	*0.7020*	0.6422	MC-OI-HOOI	**0.7605**	**0.7262**
S-TF	0.6742	0.6305	MC-MI-HOOI	0.7287	0.6756
s-Binary-Text	0.6432	0.6022	MKF	0.7327	0.6787
S-cross-citation	0.6833	0.6057	FI	0.6944	0.6031
S-co-citation	0.6815	*0.6565*	SA	0.7226	0.6952
S-Bibliographic coupling	0.4398	0.3348	AdacVote	*0.7454*	*0.7176*
S-Binary-citation	0.5831	0.5238	CP-ALS	0.7042	0.6377
			LMF	0.5935	0.5058

TABLE III

CLUSTERING PERFORMANCE ON WOS JOURNAL DATABASE

Text data	Ranking of α_i	α_i	Performance ranking
TFIDF	0.5890	1	1
IDF	0.4519	3	2
TF	0.5580	2	3
Binary-Text	0.3708	4	4
Citation	Ranking of α_i	α_i	Performance ranking
cross-citation	0.5372	2	2
co-citation	0.5771	1	1
Bibliographic coupling	0.5095	3	4
Binary-citation	0.3446	4	3

TABLE IV

THE WEIGHTING COEFFICIENTS OF MULTI-VIEW DATA OBTAINED BY MC-MI-HOOI IN JOURNAL DATA.

the intuitive confusion matrices correspond to the numerical evaluation results. For instance, the quality of clustering obtained by MC-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-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.

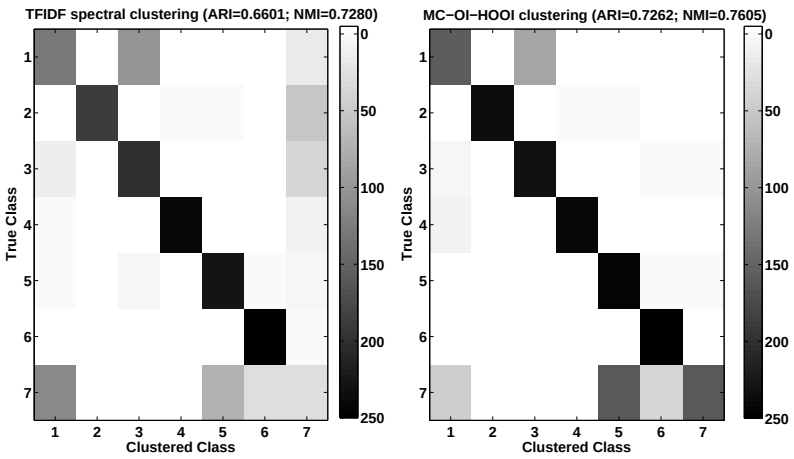

Fig. 9. Confusion matrices of journal data obtained by spectral clustering on TFIDF (left) and MC-OI-HOOI (right). The numbers of cluster labels are consistent with the numbers of ESI journal categories. In each row, the diagonal element represents the fraction of correctly clustered journals and the off-diagonal non-zero element represents the fraction of mis-clustered journals. (Data source: Thomson Reuters, Web of Science)

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 13 to Table 15. As shown, the best clustering performance is obtained by MC-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-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-MI-HOOI (NMI 0.7494, ARI 0.6015) and MC-OI-MLSVD (NMI 0.7429, ARI 0.6030). All of our tensor based methods are not only beyond spectral clustering results of any single-view data but also superior to the six baseline multi-view clustering methods, demonstrating 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. 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 Figure 10, two confusion matrices of disease gene data are depicted to illustrate the partition difference between our multi-view clustering (by MC-OI-HOOI) and the best single-view clustering result (on LDDB). The values of the matrices are normalized according to $R_{ij} = C_j/T_i$, where T_i is the total number of genes belonging to disease category i and C_j is the number of these T_i genes that are clustered to class j. In the first place, it is worth noting that MC-OI-HOOI reduces the number of mis-clustered genes for breast cancer (Nr. 1), mental retardation (Nr. 10), muscular dystrophy (Nr.11) and neuropathy (Nr. 12). Second, there are

SC-Algorithm	**NMI**	**ARI**	**MC-Algorithm**	**NMI**	**ARI**
S-GO	0.5367	0.3657	MC-OI-MLSVD	0.7429	*0.6030*
S-MeSH	*0.7072*	0.5134	MC-OI-HOOI	**0.7732**	**0.6473**
S-OMIM	0.6971	0.4901	MC-MI-HOOI	*0.7494*	0.6015
S-NCI	0.5153	0.3063	MKF	0.7002	0.5445
S-eVO	0.6048	0.3845	FI	0.6743	0.4830
S-KO	0.3187	0.1194	SA	0.7016	0.5495
S-LDDB	**0.7088**	**0.5942**	AdacVote	0.6093	0.5349
S-MP	0.6582	0.4962	CP-ALS	0.7241	0.5154
S-SNOMED	0.6819	*0.5205*	LMF	0.6058	0.4402
S-Uniprot	0.5692	0.3303			

TABLE V

CLUSTERING PERFORMANCE ON DISEASE DATA SET.

Sources	Ranking of α_i	α_i	Performance ranking
GO	9	0.2544	8
MeSH	7	0.2842	2
OMIM	4	0.2973	3
NCI	6	0.2931	9
eVO	3	0.3021	6
KO	10	0.2216	10
LDDB	1	**0.5303**	1
MP	2	*0.3113*	5
SNOMED	8	0.2713	4
Uniprot	5	0.2970	7

TABLE VI

THE WEIGHTING COEFFICIENTS OF MULTI-VIEW DATA OBTAINED BY MC-MI-HOOI IN DISEASE DATA.

several diseases where consistent mis-clustering occurs in both methods, such as, cataract (Nr. 3), charcot marie tooth disease (Nr. 4) and diabetes (Nr. 6). The intuitive confusion matrices correspond to the numerical evaluation results. As shown in Table V, the quality of clustering obtained by MC-OI-HOOI (NMI 0.7605, ARI 0.7262) is higher than that of LDDB.

In spectral clustering, by checking the "elbow" of the plot of the eigenvalues of single-view data [20], it may provide a heuristic guess on the optimal cluster number. Analogously, regarding our tensor based multi-view clustering, we also intend to explore the relationship between the

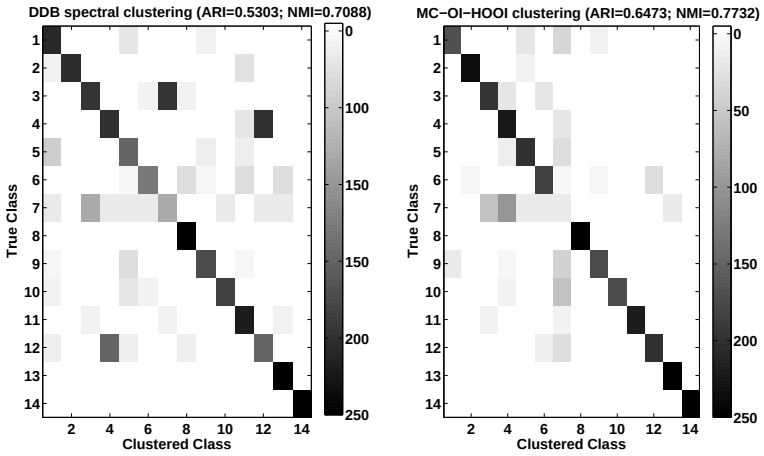

Fig. 10. Confusion matrices of disease gene data obtained by spectral clustering on LDDB (left) and MC-OI-HOOI (right). The numbers of cluster labels are consistent with the numbers of diseases. In each row, the diagonal element represents the fraction of correctly clustered genes and the off-diagonal non-zero element represents the fraction of mis-clustered genes

optimal cluster number and 1-mode singular values of similarity tensors. In Figure 11, from each tensor of our three data sets, we plot the top 20 1-mode singular values. As shown in Figure 11, the "elbow" in synthetic data is quite obvious at the number 2 or 3 (the real cluster number is 3). For journal data analysis, the "elbow" is more likely to range from 2 to 10 (the real cluster number is 7). For disease data analysis, the "elbow" is from 2 to 15 (the real cluster number is 14). It seems that checking the "elbow" of the plot of 1-mode singular value might also offer a heuristic estimation of the optimal cluster number for multi-view clustering. Moreover, in Figure 1 and Figure 2 of **Supplementary material 5**, we also compare the 1-mode singular value curves using different tensors of journal data and gene-disease data. Those tensors are generated from different numbers of views, for instance, in journal data, we generate different tensors by using various combinations from two to seven views. As shown, for each data, the 1-mode singular value plot is quite stable w.r.t. the different combinations of multiple views.

To investigate the computational time, we benchmark our tensor based multi-view clustering algorithms with 6 different multi-view clustering methods on the two application data sets. As shown in Table VII, our three tensor based strategies (MC-OI-MLSVD, MC-OI-HOOI and MC-MI-HOOI) are efficient. For instance, they are faster than four multi-view clustering methods (SA, AdacVote, CP-ALS and LMF). Obviously, MC-OI-MLSVD is more efficient. On the other hand,

although MKF and FI seem more efficient than our three tensor based algorithms, our proposed methods yield much better performance or more enriched information (the weighting factors of the single views). Meanwhile, the two clustering ensemble methods SA and AdacVote require more computation time since they involve the partition of each single-view data. Consequently, with number of views increasing, the computation of the clustering ensemble method will become more and more intensive.

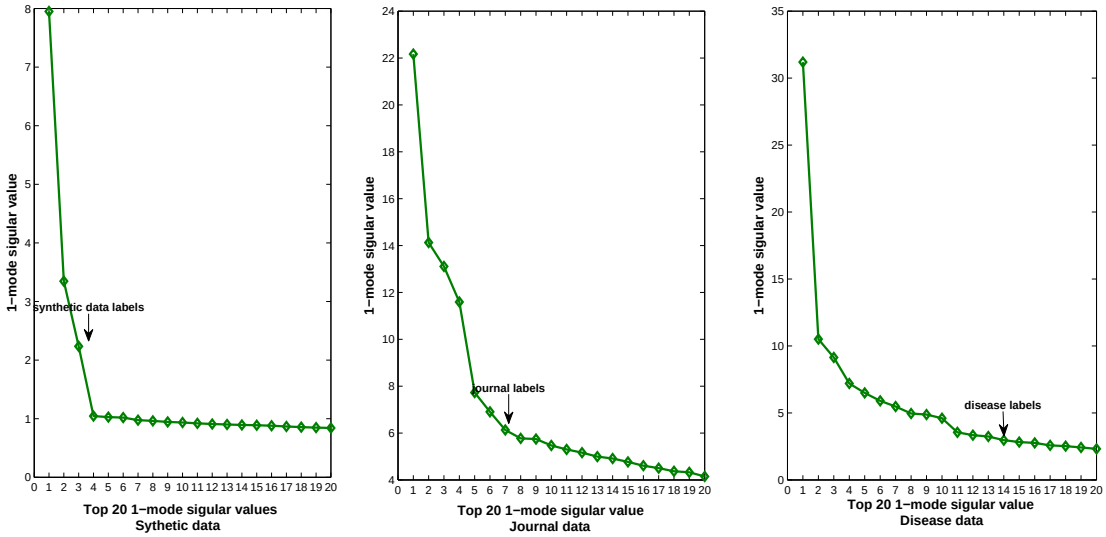

Fig. 11. Plot of the top 20 1-mode singular values of tensors constructed from different multi-view data (synthetic data on the left, journal data on the middle and disease data on the right). All possible views of each data are employed to construct these tensors.

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

Algorithm	disease data (seconds)	journal data (seconds)
MC-OI-MLSVD	4.82	33.34
MC-OI-HOOI	9.32	64.98
MC-MI-HOOI	2.79	41.75
MKF	1.23	3.97
FI	3.94	20.06
SA	37.29	60.94
AdacVote	37.31	44.67
CP-ALS	7.82	127.55
LMF	9.40	203.41

TABLE VII

COMPARISON OF CPU TIME OF ALL MULTI-VIEW CLUSTERING ALGORITHMS ON REAL APPLICATIONS

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 $\mathbf{U}$ 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 shared by all views and filter out irrelevant information or noise. The tensor based multi-view clustering of optimization integration strategy (MC-OI-MLSVD and MC-OI-HOOI) leverages the effect of each single-view data in an appropriate way while the tensor based multi-view clustering of matrix integration strategy (MC-MI-HOOI) is able to utilize the linear relationship of multi-view data for joint analysis.

One thing we need to emphasize is that, to some degree, our multi-view clustering relies on the complementary property of multi-view data. As shown in Figure 1 and Figure 8, as long as multi-view data contains enough complementary information and there is a clear cluster structure hidden amid them, our tensor based multi-view clustering methods are able to facilitate the detection of the latent cluster structure as expected. Nevertheless each single view may contain incomplete structure information or excessive noise.

In our former experiments, MC-OI-HOOI seems to work well to integrate multi-view heterogeneous data (data from different feature spaces) for joint clustering. For example, it works best to integrate ten-view data generated from different content (controlled vocabularies). Moreover, MC-OI-HOOI works best to integrate four views of text data with four views of citation data. The reason might be that MC-OI-HOOI provides an optimal common subspace shared by multi-view heterogeneous data by simultaneously analyzing them together. Meanwhile, MC-MI-HOOI appears to work well to integrate multi-view homogeneous data (data from the same feature space) for joint clustering because it utilizes the linear relationship of multi-view homogeneous data to boost the joint clustering. For example, MC-MI-HOOI works best to integrate two views of text data (TFIDF and IDF), as can be seen in Table 6 and Table 7 of **Supplementary material 2**. In addition, as shown in Table I, MC-MI-HOOI works best to integrate our multi-view synthetic data as well. In fact, our synthetic data can be regarded as a kind of homogeneous data. Thanks to its fast computation and good clustering performance, our MC-OI-MLSVD is an efficient multi-view clustering strategy for integrating multi-view heterogeneous data.

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 optimization integration (MC-OI) as well as by matrix integration (MC-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.

REFERENCES

[1] H. G. Ayad and M. S. Kamel. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):160–173, 2008.

[2] B. W. Bader and T. G. Kolda. MATLAB tensor toolbox version 2.4. http://csmr.ca.sandia.gov/ tgkolda/TensorToolbox/, March 2010.

[3] S. Bickel and T. Scheffer. Multi-view clustering. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, ICDM '04, pages 19–26, Washington, DC, USA, 2004. IEEE Computer Society.

[4] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Echart-Young', decomposition. *Psychometricka*, 35:283–319, 1970.

[5] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 129–136, New York, NY, USA, 2009. ACM.

[6] L. De Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

[7] L. De Lathauwer, B. D. Moor, and J. Vandewalle. On the best rank-1 and rank-$(r_1, r_2, \ldots, r_n)$ approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4):1324–1342, 2000.

[8] D. M. Dunlavy, T. G. Kolda, and E. Acar. Poblano v1.0: A MATLAB toolbox for gradient-based optimization. Technical Report SAND2010-1422, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, March 2010.

[9] D. M. Dunlavy, T. G. Kolda, and W. P. Kegelmeyer. Multilinear algebra for analyzing data with multiple linkages. Technical Report SAND2006-2079, Sandia National Laboratories, 2006.

[10] R. A. Harshman. Foundations of the PARAFAC procedure: Model and conditions for an 'explanatory' multi-modal factor analysis. *UCLA working papers in phonetics*, 16:1–84, 1970.

[11] H. Huang, C. Ding, D. Luo, and T. Li. Simultaneous tensor subspace selection and clustering: the equivalence of high order SVD and k -means clustering. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, pages 327–335, New York, NY, USA, 2008. ACM.

[12] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[13] M. Ishteva, L. De Lathauwer, P.-A. Absil, and S. Van Huffel. Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme. *SIAM Journal on Matrix Analysis and Applications*, 32(1):115–135, 2011.

[14] T. Joachims, N. Cristianini, and J. Shawe-Taylor. Composite kernels for hypertext categorisation. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 250–257, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[15] T. Kolda and B. Bader. The TOPHITS model for higher-order web link analysis. In *Proceedings of the SIAM Data Mining Conference Workshop on Link Analysis, Counterterrorism and Security*, 2006.

[16] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[17] P. Kroonenberg and J. de Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 1980.

[18] X. Liu, S. Yu, Y. Moreau, B. De Moor, W. Glänzel, and F. Janssens. Hybrid clustering of text mining and bibliometrics applied to journal sets. In *Proceedings of SIAM International Conference on Data Mining*, pages 49–60, Philadelphia, PA, USA, 2009. SIAM.

[19] B. Long, P. S. Yu, and Z. M. Zhang. A general model for multiple view unsupervised learning. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 822–833, 2008.

[20] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[21] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[22] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[23] M. L. Overton and R. S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. *Math. Program.*, 62(2):321–357, 1993.

[24] B. Savas and L. Eldén. Krylov-type methods for tensor computations. *arXiv:1005.0683v2 [math.NA]*, 2010.

[25] T. M. Selee, T. G. Kolda, W. P. Kegelmeyer, and J. D. Griffin. Extracting clusters from large datasets with multiple similarity measures using IMSCAND. In M. L. Parks and S. S. Collis, editors, *CSRI Summer Proceedings 2007, Technical Report SAND2007-7977, Sandia National Laboratories, Albuquerque, NM and Livermore, CA*, pages 87–103, 2007.

[26] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[27] A. Strehl and J. Ghosh. Cluster ensembles-a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[28] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, pages 374–383, New York, NY, USA, 2006. ACM.

[29] L. Tang, X. Wang, and H. Liu. Uncovering groups via heterogeneous interaction analysis. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pages 143–152, Washington, DC, USA, 2009. IEEE Computer Society.

[30] L. Tang, X. Wang, and H. Liu. Community detection in multi-dimensional networks. Technical report, School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, 2010.

[31] W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pages 1016–1021, Washington, DC, USA, 2009. IEEE Computer Society.

[32] L. Tucker. The extension of factor analysis to three-dimensional matrices. In H. Gulliksen and N. Frederiksen, editors, *Contributions to mathematical psychology*, pages 109–127. Holt, Rinehart & Winston, NY, 1964.

[33] D. Verma and M. Meila. A comparison of spectral clustering algorithms. Technical report, Department of CSE University of Washington Seattle, WA, 2003.

[34] J. Ye, R. Janardan, and Q. Li. GPCA: an efficient dimension reduction scheme for image compression and retrieval. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge Discovery and Data mining*, New York, NY, USA, 2004. ACM.

[35] D. Zhou and C. J. C. Burges. Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th international conference on Machine Learning*, pages 1159–1166, New York, NY, USA, 2007. ACM.