The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Tensor Decomposition for Multilayer Networks Clustering

Zitai Chen,[1,2] **Chuan Chen,**[1,2*] **Zibin Zheng,**[1,2] **Yi Zhu**[3]
[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[2]National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China
[3]MTdata, Meitu
chenzt25@mail2.sysu.edu.cn, {chenchuan, zhzibin}@mail.sysu.edu.cn, zhuy@meitu.com

Abstract

Clustering on multilayer networks has been shown to be a promising approach to enhance the accuracy. Various multilayer networks clustering algorithms assume all networks derive from a latent clustering structure, and jointly learn the compatible and complementary information from different networks to excavate one shared underlying structure. However, such an assumption is in conflict with many emerging real-life applications due to the existence of noisy/irrelevant networks. To address this issue, we propose Centroid-based Multilayer Network Clustering (CMNC), a novel approach which can divide irrelevant relationships into different network groups and uncover the cluster structure in each group simultaneously. The multilayer networks is represented within a unified tensor framework for simultaneously capturing multiple types of relationships between a set of entities. By imposing the rank-$(L_r, L_r, 1)$ block term decomposition with nonnegativity, we are able to have well interpretations on the multiple clustering results based on graph cut theory. Numerically, we transform this tensor decomposition problem to an unconstrained optimization, thus can solve it efficiently under the nonlinear least squares (NLS) framework. Extensive experimental results on synthetic and real-world datasets show the effectiveness and robustness of our method against noise and irrelevant data.

1 Introduction

Many real-world networks and complex system are represented as a set of entities interacting with each other via multiple types of relationships (Wasserman and Faust 1994; Kivelä et al. 2014). Since different networks have different data distributions, it is reasonable to separate them into several homogeneous networks to utilize the traditional tools in graph (Cai et al. 2005). For example, in aeronautical flight system (Cardillo et al. 2013), different airports (cities) are connected by flight routes constituted by diverse airlines networks; in gene co-expression networks (Ficklin and Feltus 2011), genes often play different roles in different tissues. From the viewpoint of each individual network, they respectively contain a partial description of the whole system and might be no sufficient to accomplish a learning task alone for its incompleteness and noise. Thus, mining on multilayer networks becomes a crucial task to improve our understanding of the integral system.

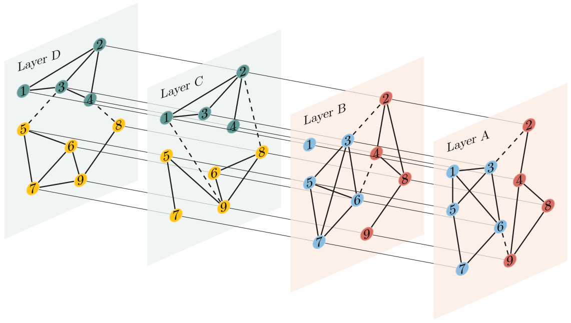


Figure 1: An example of Multilayer Networks: the network system is comprised by 4 *networks* $\{A, B, C, D\}$ and 9 nodes. The informative edges are represented in solid line while noisy edges in dashed line. There are 2 *network groups* $C_1^{Net} = \{A, B\}$ and $C_2^{Net} = \{C, D\}$, each of which has 2 *node clusters* with same color: $C_1^{Node} = \{C_{Blue}^{Node}, C_{Red}^{Node}\}$ and $C_2^{Node} = \{C_{Green}^{Node}, C_{Yellow}^{Node}\}$. The unobserved node v_1 in $\{B\}$ is not linked by any of other nodes, which raises a challenge of incompleteness.

As a typical unsupervised learning problem, clustering aims to find a set of samples closed to each other in the same cluster and far away from different clusters. It has been widely studied on both feature-based and graph-based learning. One of the most adopted feature-based methods is K-means algorithm (Lloyd 1982), which groups a given dataset into k clusters by optimizing the cost function of the total distance from points to cluster centroids. On network cluster learning, spectral clustering (von Luxburg 2007) has attracted much attention for the good performance. However, it also suffers from the sensitiveness, post-processing and difficulty in interpretation. The symmetric nonnegative matrix factorization (Kuang, Park, and Ding 2012) has been shown to be a promising approach giving well interpretable and comparative results. It approximates the adjacency matrix $\boldsymbol{X}$ with a nonnegative low-rank factor, i.e. $\boldsymbol{X} \approx \boldsymbol{H}\boldsymbol{H}^T, (\boldsymbol{H} \geqslant 0)$, in which clustering assignment of each data can be easily obtained by finding the largest entry in the corresponding row of $\boldsymbol{H}$.

[*]Corresponding author

For multilayer networks clustering problem, the key assumption of the existing multilayer networks algorithms is that all networks share one underlying clustering structure (Dunlavy, Kolda, and Kegelmeyer 2011; Kolda, Bader, and Kenny 2005). By leveraging the dependency, coherence and complementarity of networks, multilayer networks clustering is able to provide better performance. As an illustration, in Figure 1, although the two-layer networks $\{A, B\}$ has different linkages between the same set of nodes, it can learn a unique clustering structure from the compatible information. With the complementary information from network $\{A\}$, the unobserved node $v_1 \in \{B\}$ has higher probability to be clustered to C^{Node}_{Blue} rather than C^{Node}_{Red}. And the distraction from noisy edges could also be alleviated.

However, this assumption does not always hold in real life, and a noisy/irrelevant network can dramatically deviate the result from the real one. A widely used solution is to lower the importance of the noisy networks in learning procedure, which needs a time-consuming parameter tuning. As shown in Figure 1, the tuning on importance face a dilemma when another group of networks $\{C, D\}$ exists. Moreover, excluding networks mistakenly might miss an opportunity to uncover another valuable data distribution covered in the multilayer networks. Taking the user networks as an example of multilayer networks, different functional applications have their meaningful relationship across users respectively, such as *Yelp* suggesting the flavor preference, while *Youtube* and *Netflix* revealing the entertainment inclination. The various networks could follow completely different underlying cliques, while the similar functional networks could provide complementary information like social networks *Facebook* and *Twitter*. That is key knowledge manifested in 4 networks of Figure 1.

Thus, it is necessary and realistic to obtain a macroscopic view by grouping networks, and learn a micro-structure by clustering nodes in each network group. In this paper, we propose a tensor decomposition based clustering algorithm CMNC to succeed in clustering networks and nodes simultaneously. Our contributions are summarized as follows:

- We propose a realistic clustering problem: clustering intra-layer networks and inter-layer network. A comprehensive description of the node can be acquired by the various clusterings among different networks.
- We develop an interpretable and well-performed clustering model: Centroid-based Multilayer Network Clustering. Within the tensor framework, the multilayer networks could be modeled across networks.
- CMNC is hyperparameter free, which is a major advantage in unsupervised learning. Numerically, by introducing 2 operators, the constrained optimization problem can be efficiently solved in an unconstrained NLS framework.
- Extensive experiments are conducted to verify the effectiveness and robustness of the proposed method.

The rest of the paper is organized as follows: A brief overview of clustering methods is provided, followed by some preliminaries, the proposed CMNC model and optimization; Then experimental results are presented; Finally conclusion and future works are discussed.

2 Related work

Single source data clustering The clustering method can be mainly classified into three scenarios: feature-based, graph-based, and hybrid clustering method. The well-known feature-based clustering methods include the K-means algorithm, hierarchical clustering (Newman 2004) and NMF (Paatero and Tapper 1994) methods. By transforming to a similarity graph cut problem, many graph clustering algorithms have been proposed, such as spectral clustering (von Luxburg 2007), SymNMF (Kuang, Park, and Ding 2012) and modularity maximization (Newman 2006). For the hybrid clustering method, (Cai et al. 2011) proposes an NMF with graph regularization and (Du, Drake, and Park 2017) proposes a joint NMF and SymNMF clustering framework .

Multilayer networks clustering The prior research on node-aligned multilayer networks clustering is mostly proposed for multi-view clustering and multilayer graphs. These methods can be regarded as the extension of the single source clustering method, such as the extension of NMF (Liu et al. 2013a; Li, Jiang, and Zhou 2014), spectral clustering (Kumar and III 2011) and modularity maximization (Didier, Brun, and Baudot 2015). For instance, (Tang, Lu, and Dhillon 2009) proposes linked matrix factorization(LMF) to link adjacency matrices by a shared factor $\boldsymbol{H}$: $\boldsymbol{X}^{(m)} \approx \boldsymbol{H}\boldsymbol{\Lambda}^{(m)}\boldsymbol{H}^T$. Similarly, (Dong et al. 2012) proposes an eigen-decomposition method to approximate the graph Laplacian: $\boldsymbol{L}^{(m)} \approx \boldsymbol{P}\boldsymbol{\Lambda}^{(m)}\boldsymbol{P}^{(-1)}$. (Kumar, Rai, and Daume 2011) proposes a joint-spectral clustering with a co-regularized term. Similarly, (Liu et al. 2013a) proposed a joint-NMF framework with a common consensus regularization.

Nevertheless, multiple matrices representation interrupt the analysis of a node across different views, which is non-negligible in learning both the networks and the clusterings. To preserve the view factor and take the multilayer networks as a whole, tensor representation gives a helpful resolution. The adjacency tensor can preserve the network structure as well as provide a lot of tensor-decomposition tools for analysis (Kolda and Bader 2009; Cichocki et al. 2015; Vervliet, Debals, and Lathauwer 2016; Chen et al. 2018). (Liu et al. 2013b) develops a tensor-based framework of multi-view spectral clustering by MLSVD. (Dunlavy, Kolda, and Kegelmeyer 2011) utilizes the CP decomposition to analyze the multi-link graphs.

However, all these methods assume various networks share one underlying clustering structure, and it is not easy for them to accommodate the multi-structure tasks. Recently, some works on multiple structure clustering have been proposed in a related field multi-domain clustering, which models the inner- and cross-domain linkages in clusters (Ni et al. 2015). These methods need an additional relationship between networks in the analysis, which is a kind of guidance on network clustering.

3 Preliminaries

Problem Definition A multilayer networks is defined as $\mathcal{G} = \langle V, \mathcal{E} \rangle$, where $V = \{v_i\}_{i=1}^{n}$ is a set of n nodes and

$\mathcal{E} = \{E_i\}_{i=1}^N$ is a set of relationships. The multilayer networks can be separated into N relatively independent of networks $\{G_i\}_{i=1}^N$, where $G_i = \langle V, E_i \rangle$, and represented by the adjacency matrices $\{\boldsymbol{X}_i\}_{i=1}^N$.

Formally, multilayer networks clustering aims to partition N networks into R network groups $C^{Net} = \{C_i^{Net}\}_{i=1}^R$, where $C_r^{Net} = \{G_{r_1}, \cdots, G_{r_{l_r}}\}$ is the r^{th} group containing l_r graphs. Meanwhile, it learns node clusterings $C_r^{Node} = \{C_{r,1}^{Node}, \cdots, C_{r,L_r}^{Node}\}$ from the r^{th} network group, where $C_{r,i}^{Node}$ is the i^{th} node cluster in C_r^{Net} and L_r is the number of node clusters in r^{th} network cluster. We suppose that the cluster number R and $L_r, (r = 1, \cdots, R)$ are given.

Notation and Preliminaries Tensor is a multidimensional array. Let $\boldsymbol{\mathcal{X}}$ be an m-order tensor of size $I_1 \times I_2 \times \cdots \times I_m$. *Rank-one* tensor can be written as the outer product of m vectors, i.e. $\boldsymbol{\mathcal{X}} = \mathbf{x}^{(1)} \circ \cdots \circ \mathbf{x}^{(m)}$. The *mode-p* matricization of $\boldsymbol{\mathcal{X}}$ is denoted as an $I_p \times (I_1 \cdots I_{p-1} I_{p+1} \cdots I_m)$ matrix $\boldsymbol{\mathcal{X}}_{(p)}$, which is obtained by the rearrangement of element. The vectorization of $\boldsymbol{\mathcal{X}}$ is denoted as $vec(\boldsymbol{\mathcal{X}})$. The Frobenius norm $\|\boldsymbol{\mathcal{X}}\|_F$ is the sum of the squares of all its elements $a_{i_1 i_2 \ldots i_m}$: $vec(\boldsymbol{\mathcal{X}})^T vec(\boldsymbol{\mathcal{X}})$. The *Khatri-Rao* product is denoted as $\odot$. The $n \times n$ identity matrix is denoted by $\boldsymbol{I}_n$ and the all-one vector by $\mathbf{1}_{n \times 1}$ or $\mathbf{1}_n$ for short. Matrix is denoted as bold uppercase letter $\boldsymbol{A}$ and its elements in lowercase a_{ij}. $\boldsymbol{A}_{i:}$ and $\boldsymbol{A}_{:j}$ denote the i^{th} row and j^{th} column of $\boldsymbol{A}$.

The multilinear rank-$(L_r, L_r, 1)$ terms decomposition (Lathauwer 2008), a tensor decomposition format, is applied as a foundational model in this paper. It writes a third-order tensor as a sum of R low multilinear rank terms, each of which can be written as the outer product of a rank-L_r matrix and a vector, i.e. $\boldsymbol{\mathcal{X}} = \sum_{r=1}^R (\boldsymbol{A}_r \boldsymbol{B}_r^T) \circ \mathbf{c}_r$., where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$, and $\boldsymbol{A}_r \in \mathbb{R}^{I \times L_r}$ $\boldsymbol{B}_r \in \mathbb{R}^{J \times L_r}$ and $\mathbf{c}_r \in \mathbb{R}^{K \times 1}$. We denote this form of decomposition as $\boldsymbol{\mathcal{M}}_{LL1}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$, where $\boldsymbol{A}$ is the concatenation of matrix $[\boldsymbol{A}_1 \cdots \boldsymbol{A}_R] \in \mathbb{R}^{I \times R'}$ with $R' = \sum_{r=1}^R L_r$, so do $\boldsymbol{B}$ and $\boldsymbol{C}$. Solving the exact decomposition is NP-hard (Kolda and Bader 2009), and it resorts to the approximation as follow:

$$\min_{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}} \quad \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{M}}_{LL1}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})\|_F^2. \tag{1}$$

4 The Proposed Method

In this section, we focus on multilayer networks clustering and describe the proposed model (i.e. CMNC). We start with model details, followed by discussion and analysis.

Centroid-based Multilayer Networks Clustering

In this problem setting, there exists two aspects of clustering, the networks and the nodes, both of which are of importance. A good network grouping provides fertile ground for node clustering, and the better clustering results of each network group can be found. Inspired by multi-view methods, the initial idea is to apply a two-step strategy: grouping networks first and then learn a multi-view task. We explain the key idea in the following simplified case under the tensor representation framework.

The Simplified Case Suppose $\boldsymbol{X}_i \in \mathbb{R}^{n \times n}$ is the adjacency matrix of the i^{th} network G_i with n nodes. To maintain the structure of multilayer networks, N adjacency matrices are aligned and compacted along the third mode of a third-order tensor, namely $\boldsymbol{\mathcal{X}}_{:,:,i} = \boldsymbol{X}_i, i = 1, \cdots, N$.

With the tensor representation, seeking a partition of the networks is equivalent to finding a sum of source component tensors to restore the adjacency tensor, and networks in such component share the same underlying node structure. As aforementioned, the rank-$(L_r, L_r, 1)$ terms decomposition meets the need of partition, which recovers the component tensors by duplicating the basic matrix $(\boldsymbol{A}_r \boldsymbol{B}_r^T)$ by multiplying different weights. It can also be interpreted as generating the same structure networks of various weights. We suppose that each network only belongs to a unique component tensor, so-called hard clustering on networks. It is natural to set the component's weight vectors $\mathbf{c}_r$ in $\boldsymbol{\mathcal{M}}_{LL1}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ as an indicator to partition networks into different component tensors, namely

$$\mathbf{c}_r = [0, \cdots, 0, 1, \cdots, 1, 0, \cdots, 0]^T \tag{2}$$

pointing out the tied networks with 1 and ignoring the irrelevant networks with 0 in the r^{th} component. And each network ought to be indicated only once. Formally, each row of the concatenated matrix of indicators $\boldsymbol{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_R]$ is the one-hot vector. We can adjust the decomposition with the indicator $\boldsymbol{C}$ as follow:

$$\begin{aligned} \min \quad & \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{M}}_{LL1}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{C} \in \{0, 1\}, \quad \boldsymbol{C}\mathbf{1}_R = \mathbf{1}_N. \end{aligned} \tag{3}$$

Owing to the nonnegative and symmetric adjacency matrices $\boldsymbol{X}_i$, we adopt the SymNMF (Kuang, Park, and Ding 2012) as the decomposition format by imposing $\boldsymbol{A}_r = \boldsymbol{B}_r$ to the basic matrix. To sum up, we can formulate the objective function of the simplified case as follow:

$$\begin{aligned} \min \quad & \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{M}}_{LL1}(\boldsymbol{A}, \boldsymbol{A}, \boldsymbol{C})\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{A} \geqslant 0, \quad \boldsymbol{C} \in \{0, 1\}, \quad \boldsymbol{C}\mathbf{1}_R = \mathbf{1}_N. \end{aligned} \tag{4}$$

The networks cluster is obtained by the indicator $\boldsymbol{C}$ while the nodes cluster is obtained by the largest entry in the corresponding row of $\boldsymbol{A}_i$. In other word, if $k = \arg\max_j \{(\boldsymbol{A}_i)_{lj}\}$, then node $v_l \in C_i^{Net}$ belongs to k^{th} node cluster: $v_l \in C_{i,k}^{Node}$.

The General Case As we can see, the assumption of hard clustering is too strict to hold in general. A multilayer networks system usually evolved out of mixing multiple weighted network components together. Dropping out the assumption made before, it is easy to extend the simplified case to this general case by relaxing the discrete value constraint to positive continuous value on $\boldsymbol{C}$, so-called fuzzy clustering. So we present CMNC as:

$$\begin{aligned} \min \quad & \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{M}}_{LL1}(\boldsymbol{A}, \boldsymbol{A}, \boldsymbol{C})\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{A} \geqslant 0, \quad \boldsymbol{C} \geqslant 0, \quad \boldsymbol{C}\mathbf{1}_R = \mathbf{1}_N. \end{aligned} \tag{5}$$

The networks group is also obtained by the largest entry in the corresponding row of $\boldsymbol{C}$, namely $G_i \in C_k^{Net}$ if and only if $k = \arg\max_j \{c_{ij}\}$. Now, the assignment of the network cluster can be interpreted as a probability distribution among all the available structures.

Discussion We claim that CMNC can be seen as a structured K-means algorithm, and that's why it is named as centroid-based. Indeed, the resulting basic network of each component is the centroid of all the networks contained. The resulting basic network of each component does guide the node clustering by setting itself as the target network.

With *mode*-3 matricization of tensor $\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{M}}_{LL1}$, it is more clear to explicate the essence of CMNC under the matrix framework:

$$\begin{aligned} \min \quad & \|\boldsymbol{\mathcal{X}}_{(3)} - \boldsymbol{C}\boldsymbol{S}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{A}, \boldsymbol{C} \geqslant 0, \quad \boldsymbol{C}\mathbf{1}_R = \mathbf{1}_N, \\ & \boldsymbol{S} = [vec(\boldsymbol{E}_1) \cdots vec(\boldsymbol{E}_R)]^T, \\ & \boldsymbol{E}_r = \boldsymbol{A}_r \boldsymbol{A}_r^T, \quad r = 1, \cdots, R, \end{aligned} \tag{6}$$

where $\boldsymbol{\mathcal{X}}_{(3)} \in \mathbb{R}^{N \times (n^2)}$, $\boldsymbol{S} \in \mathbb{R}^{R \times (n^2)}$. Note that the K-means method has the following matrix form:

$$\begin{aligned} \min \quad & \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{M}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{Z} \in \{0, 1\}, \quad \boldsymbol{Z}\mathbf{1}_R = \mathbf{1}_N, \end{aligned} \tag{7}$$

where $\boldsymbol{M}$ is the clustering centroid and $\boldsymbol{Z}$ is the cluster indicator. Compared to Eq. (7), Eq. (6) is entirely the matrix form of K-means with additional constraints on the centroid matrix $\boldsymbol{S}$, the vectorized adjacency matrices. In the simplified case, each network group's centroid is the means of the networks in the same cluster, where the objective function can be formulated as

$$\min \sum_{r=1}^{R} \| \sum_{i \in \mathbf{c}_r} \boldsymbol{X}_i - \|\mathbf{c}_r\| \cdot vec(\boldsymbol{A}_i \boldsymbol{A}_i^T) \|_F^2, \tag{8}$$

which is K-means with non-negative positive definite symmetric constraint on factor (centroid) matrix. It is also established in general case with more complicated explanation and the stronger ability of expressions.

Since K-means assumes that the data points in each cluster follow a spherical Gaussian distribution, it is also assumed that the networks in the same network group are generated from an underlying structure with additive noise following the same distribution, which vitalizes centroid network representation in CMNC. We also normalize the adjacency matrices ($\boldsymbol{X}_i := \boldsymbol{D}^{-1/2}\boldsymbol{X}_i\boldsymbol{D}^{-1/2}$, where $\boldsymbol{D}$ is the degree matrix) to avoid the side effect of the scaling of adjacency matrix in weighted case. This normalization procedure is also important in the proof of graph cut problem.

The difference between the 2 step clustering of K-means with SymNMF and CMNC is mainly in twofold: on the one hand, the vectorized adjacency matrix is in an extremely high dimension which will suffer from the so-called "curse of dimensionality"; on the other hand, the few network samples in K-means will be unstable in cluster, while the tensor decomposition iteratively updating the twofold clustering could mostly avoid the mistake from one shot decision.

Extension for Incomplete Network In multilayer networks, that a node disappeared in some layers is a universal phenomenon. To tackle this issue, we need a so-called observation tensor $\boldsymbol{\mathcal{W}}$ of the same size with ones in the positions corresponding to known entries of the dataset and zeros elsewhere. That's to say when node i disappeared in network j, then $\boldsymbol{\mathcal{W}}(i, :, j)$ and $\boldsymbol{\mathcal{W}}(:, i, j)$ will be all zeros. Remaining the constraints unchanged, the objective is:

$$\min \quad \|\boldsymbol{\mathcal{W}} * (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{M}}_{LL1}(\boldsymbol{A}, \boldsymbol{A}, \boldsymbol{C}))\|.$$

Relationship with Graph Cut

In this section, we prove that CMNC is a relaxation of the normalized multilayer networks cut problem. The normalized graph cut (Shi and Malik 2000) on a network G is to minimize the loss: Cut(X), and the solution is addressed by:

$$\boldsymbol{A} = \arg \min_{\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}, \boldsymbol{A} \geqslant 0} \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{A}^T\|^2, \tag{9}$$

where X is normalized adjacency matrix. Owning to multiple structures contained multilayer networks clustering, multiple graph cut patterns $\{\text{Cut}_i(G_j)\}_{i=1}^R$ are needed. The multilayer networks normalized cut is to find the minimum cut pattern for each network G_j in R available cut patterns:

$$\begin{aligned} & \min_{\text{Cut}_i} \sum_{j=1}^{n} \min\{\text{Cut}_1(G_j), \cdots, \text{Cut}_R(G_j)\} \\ = & \min_{\text{Cut}_i, \boldsymbol{C}} \sum_{j=1}^{n} \sum_{i=1}^{R} c_{ij} \text{Cut}_i(G_j) \\ = & \min_{\boldsymbol{A}_i^T\boldsymbol{A}_i = \boldsymbol{I}, \boldsymbol{A}_i \geq 0, \boldsymbol{C}} \sum_{j=1}^{n} \|\boldsymbol{X}_j - \sum_{i=1}^{R} c_{ij} \boldsymbol{A}_i \boldsymbol{A}_i^T\|^2. \end{aligned} \tag{10}$$

where $\boldsymbol{C} \in \{0, 1\}$ and $c_{ij} = 1$ only when the Cut$_i$ is minimum among all Cut on graph G_j.

In short, CMNC is the multilayer networks normalized cut on relaxing the orthogonal constraints on $\boldsymbol{A}_i$.

5 Optimization

In this section, we develop our constrained problem to an equivalent unconstrained optimization problem which can be efficiently solved with nonlinear least squares approach.

Problem Transformation

We remark the problem with new notations:

$$\begin{aligned} & \min f(\mathbf{z}) = \frac{1}{2}\|\boldsymbol{\mathcal{F}}\|_F^2, \\ & \text{s.t.} \quad \tilde{\boldsymbol{A}}, \tilde{\boldsymbol{C}} \geqslant 0, \quad \tilde{\boldsymbol{C}}\mathbf{1}_R = \mathbf{1}_N. \end{aligned} \tag{11}$$

where the residual tensor $\tilde{\boldsymbol{\mathcal{F}}} = \boldsymbol{\mathcal{M}}_{LL1}(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{A}}, \tilde{\boldsymbol{C}}) - \boldsymbol{\mathcal{X}}$ and variables $\mathbf{z} = [vec(\tilde{\boldsymbol{A}})^T; vec(\tilde{\boldsymbol{C}})^T]^T$.

We introduce two continuously differentiable operators to avoid the constraints on variables $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{C}}$. They are element-wise square operator $[\cdot]^2$ and the row-wise normalization operator$[\cdot]^{rn}$:

$$[\tilde{\boldsymbol{A}}]^2 = \tilde{\boldsymbol{A}} * \tilde{\boldsymbol{A}}, \quad [\tilde{\boldsymbol{C}}]^{rn} = [\tilde{\boldsymbol{C}}_{i:}/\|\tilde{\boldsymbol{C}}_{i:}\|]_{i=1}^{N}. \tag{12}$$

In this way, we can rewrite problem (11) as follow

$$\min f(\mathbf{z}) = \frac{1}{2}\|\boldsymbol{\mathcal{F}}\|_F^2, \tag{13}$$

where $\boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{M}}_{LL1}(\boldsymbol{A}, \boldsymbol{A}, \boldsymbol{C}) - \boldsymbol{\mathcal{X}}, \boldsymbol{A} = [\tilde{\boldsymbol{A}}]^2, \boldsymbol{C} = [[\tilde{\boldsymbol{C}}]^{rn}]^2$, the variables $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{C}}$ are unconstrained and the intermediate variables $\boldsymbol{A}$ and $\boldsymbol{C}$ satisfy the constraints.

Algorithm 1: Framework of the trust region method.

```
Input: $\mathbf{z}_0, \Delta > 0, \epsilon > 0, k \triangleq 0$;
while not Convergent do
    Solve problem (15) with Algorithm 2 for $\mathbf{p}_k^*$;
    $\gamma_k = \frac{f(\mathbf{z}_k) - f(\mathbf{z}_k + \mathbf{p}_k^*)}{m_k^f(0) - m_k^f(\mathbf{p}_k^*)}$;
    Update
    $\Delta_{k+1} \triangleq \begin{cases} 2\Delta_k & \gamma_k > 0.75 \& \|\mathbf{p}_k^*\| = \Delta_k, \\ \Delta_k/4 & \gamma_k < 0.25, \\ \Delta_k & \text{otherwise;} \end{cases}$;
    Update $\mathbf{z}_{k+1} \triangleq \begin{cases} \mathbf{z}_k, & \gamma_k \leqslant 0, \\ \mathbf{z}_k + \mathbf{p}_k^*, & \text{otherwise;} \end{cases}$;
    $k \triangleq k + 1$;
end
return $\mathbf{z}_k$.
```

Optimization Framework

Trust Region Method To optimize the unconstrained nonlinear least squares (NLS) problem (13), we adopt the trust region method (Wright and Nocedal 2006) which is presented in **Algorithm 1**. We iteratively improve an initial solution $\mathbf{z}_0$ with additive updates $\mathbf{p}_k^*$ obtained by minimizing a second-order approximation of objective function based solely on first-order derivatives. Thus, the solution of NLS problem (13) can be replaced by the solution of a sequence of the trust region subproblem of updating step $\mathbf{p}_k^*$.

In detail, given an objective f and a current solution $\mathbf{z}_k$, by linearizing the residual tensor $\boldsymbol{\mathcal{F}}(\mathbf{z}_k + \mathbf{p})$ with $m_k^{\boldsymbol{\mathcal{F}}}$, the NLS is approximated by the linear least squares problem:

$$\min_{\mathbf{p}} \quad m_k^f(\mathbf{p}) \triangleq \frac{1}{2}\|m_k^{\boldsymbol{\mathcal{F}}}(\mathbf{p})\|^2, \tag{14}$$

where $m_k^{\boldsymbol{\mathcal{F}}}(\mathbf{p}) \triangleq \boldsymbol{\mathcal{F}}(\mathbf{z}_k) + \boldsymbol{J}_k\mathbf{p}$, $m_k^f(\mathbf{p})$ is kind of second-order approximation of $f(\mathbf{z}_k)$ and Jacobian $\boldsymbol{J}_k = \partial vec(\boldsymbol{\mathcal{F}})/\partial \mathbf{z}^T$ can be computed by chain rule with the differentiable operators. Thus the trust region subproblem is:

$$\begin{aligned} \min_{\mathbf{p}} \quad & m_k^f(\mathbf{p}) \triangleq \frac{1}{2}\|\boldsymbol{\mathcal{F}}(\mathbf{z}_k) + \boldsymbol{J}_k\mathbf{p}\|^2 \\ \text{s.t.} \quad & \|\mathbf{p}_k\| \leqslant \Delta_k, \Delta_k > 0, \end{aligned} \tag{15}$$

where Δ_k is the trust region radius. After solving the approximated updating step problem within the trust region, we evaluate the approximation's validity with γ_k, the ratio of actual reduction and predicted reduction. Then the trust region radius Δ_k is updated according to how good the approximation is. The thresholds 0.75 and 0.25 for γ_k are common settings. Finally, the next iteration point $\mathbf{z}_{k+1}$ will be updated if a descending step is given. Until now, the remaining problem is how to solve the subproblem (15) effectively.

Dogleg Approach In trust region method, the Dogleg algorithm (Wright and Nocedal 2006) is widely adopted to compute the updating subproblem (15) of search step $\mathbf{p}_k^*$ combining with the Gauss-Newton step $\mathbf{p}_k^{GN}$ and the steepest descent step $\mathbf{p}_k^{SD}$. The framework of Dogleg approach is presented in **Algorithm 2**. The gradient is $g(\mathbf{z}) = df/d\mathbf{z}$.

Algorithm 2: Framework of Dogleg approach.

```
Input: $\Delta_k > 0$, Jacobian $\boldsymbol{J}_k$, Residual $\boldsymbol{\mathcal{F}}(\mathbf{z}_k)$
Calculate $\mathbf{p}_k^{SD} = -g(\mathbf{z}_k)$;
Calculate $\mathbf{p}_k^{GN} = -(\boldsymbol{J}_k^T\boldsymbol{J}_k)^{-1}g(\mathbf{z}_k)$;
Solve $\mathbf{p}_k^*$
    $\triangleq \begin{cases} \mathbf{p}_k^{GN} & \|\mathbf{p}_k^{GN}\| \leqslant \Delta_k, \\ \frac{\Delta_k}{\|\mathbf{p}_k^{SD}\|}\mathbf{p}_k^{SD} & \alpha_k\|\mathbf{p}_k^{SD}\| \geqslant \Delta_k, \\ (1-\beta)\alpha_k\mathbf{p}_k^{SD} + \beta\mathbf{p}_k^{GN} & \text{otherwise,} \end{cases}$
    where $\alpha_k = \frac{\|\mathbf{p}_k^{SD}\|^2}{\|\boldsymbol{J}_k\mathbf{p}_k^{SD}\|^2}$ and $\beta$ s.t. $\|\mathbf{p}_k^*\| = \Delta_k$;
return $\mathbf{p}_k^*$.
```

After calculating the steepest descent direction $\mathbf{p}_k^{SD}$ and the Gauss-Newton step $\mathbf{p}_k^{GN}$, we try to find a trade-off between them within the trust region Δ_k.

6 Experiment

In this section, we conduct experiments to validate the robustness and effectiveness of CMNC against noise and multiple structures. We evaluate CMNC with 6 baseline methods on synthetic dataset, 20 Newsgroups and Digits dataset.

Comparison Methods

In this part, we introduce the baseline methods in three categories: pure network clustering, multi-view clustering and multilayer networks with multiple structures. In particular,

SymNMF (Kuang, Park, and Ding 2012) performs a nonnegative symmetric factorization on each similarity matrix, which captures the cluster structure in the representation.

SC (von Luxburg 2007) analyzes the graph spectrum and learns eigenvector-based solutions. We adopted K-means as the post-processing method.

CTSC (Kumar and III 2011) incorporate spectral clustering with the co-training strategy, which is widely used in semi-supervised learning.

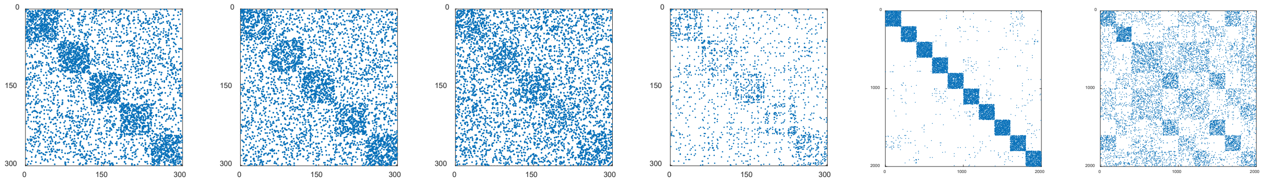
PairCRSC & CentCRSC (Kumar, Rai, and Daume 2011) adopts co-regularization framework to **SC**. The views' importance hyperparameters are set as suggested.

NONCLUS (Ni et al. 2015) is a multi-domain method clustering on domains' network and nodes' network. We construct the domain network with a clear clustering structure, in which edges only exist inside the group of networks.

SymNMF and SC are pure network clustering method for evaluating the clustering property of each network. They run the experiments network by network. CTSC, PairCRSC and CentCRSC are baseline multi-view methods. They run the experiments on the network groups separately to evaluate how rich information the complementary networks contains. Some of these methods offer an individual result for each network, while CMNC can learn complementary networks into a unique result.

Synthetic Dataset

We construct synthetic data of complete multilayer networks (**Comp**) and the incomplete one (**Incomp**). Networks in the



(a) G_1 $\in$**Comp** (b) G_3 $\in$**Comp** (c) G_6 $\in$**Comp** (d) G_1 $\in$**Incomp** (e) Pixel (f) Histogram

Figure 2: Examples of adjacency matrix, 2a, 2b and 2c are in **Comp**, 2d is in **Incomp**, 2e and 2f are in **Digits**.

Table 1: Average NMI on synthetic networks **Comp**.

Method	C_1^{Net}		C_2^{Net}			C_3^{Net}			
	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9
SymNMF	0.7393	0.6611	0.3846	0.3763	0.2659	0.1518	0.0703	0.0849	0.0496
SC	0.7537	0.6985	0.4034	0.3589	0.2611	0.1109	0.0602	0.0738	0.0647
CTSC	0.9169	0.9246	0.9329	0.9293	0.8943	0.4898	0.4201	0.5644	0.4823
PairCRSC	0.9327	0.9272	0.9320	0.9410	0.8728	0.3989	0.3655	0.4223	0.3764
CentCRSC	0.8970	0.8876	0.8868	0.8241	0.8041	0.2675	0.2495	0.2660	0.2259
NoNCLUS	0.8381	0.7661	0.7432	0.7208	0.6818	0.1649	0.0926	0.1211	0.0793
CMNC	**0.9712**		**0.9684**			**0.6588**			

same group are derived from the same structure and the difference are randomly built. To simulate different structures, we reshuffle nodes order in adjacency matrix.

Comp: We construct the $R = 3$ groups of networks with $n = 300$ nodes, where each group has 2, 3, 4 networks respectively and each underlying structure has $L_r = 5$ nodes' clusters. For generating networks, we randomly sample edges with probability α within each cluster, while with probability β from all the network to simulate noisy edges. The adjacency matrix set to 1 if edge exists, otherwise 0. Keeping the sparsity around 5%, we sample edges with ascending signal-noise ratio by tuning pair (α, β) from (0.1,0.03), (0.08,0.034) to (0.05,0.04). And the adjacency matrices before reshuffling are shown in Figure 2.

Incomp: We generate the adjacency tensor with the same setting as **Comp** except for the networks number with 5, 6, 7. To simulate the unobserved points in the networks, we randomly set the corresponding row and column of observation tensor $\boldsymbol{\mathcal{W}}$ to all zeros. The percentage of unobserved node follows a $\mathcal{N}(0.3, 0.05)$ Gaussian distribution.

In experiments, we run 100 times for each method and the average Normalized Mutual Information(NMI), the higher the better, is adopted to evaluate the performance. The results of **Comp** are shown in Table 1. Compared to the single network method, the multi-view, multi-domain and our method benefit from the complementary networks. The stable performances in different structures show the robustness of our method. Moreover, our method is not only producing a better performance than the other baseline methods but also capable of automatically grouping the relevant networks together rather than separate them with the prior knowledge. The unique clustering assignment in our model gives a clear instruction in real life application.

The result of **Incomp** is shown in Table 2. Since the spectral methods can not deal with the incomplete network and NoNCLUS ignore the unobserved node in its result, we compare our method to NMF only. With the same set of parameter, networks losing 30% information extremely decrease the performance of NMF from about 0.73 to 0.28. While our method can still maintain the performance with more networks, even none of them can provided the complete information of the network's structure.

Table 2: Average NMI on synthetic networks **Incomp**.

Method	C_1^{Net}	C_2^{Net}	C_3^{Net}
	$G_1 \sim G_5$	$G_6 \sim G_{11}$	$G_{12} \sim G_{18}$
SymNMF	0.2858	0.1213	0.0378
CMNC	**0.9344**	**0.9292**	**0.6020**

20 Newsgroups (20-NG)

We further evaluate the effectiveness of our methods using 20-Newsgroups dataset (term × document frequency), which is organized into 20 different topics. The similarity of two documents is computed by the cosine similarity of their tf-idf, which reflect the importance of each term to a document. We construct the weighted graphs by the 10-nearest-neighborhood according to the similarity.

We use 12 news group of three categories including *Comp*, *Rec* and *Talk*. From each category, we generate 5 graphs with 4 contained topics corresponding to 4 clusters in this group. We randomly sample 200 documents from 4 topics (50 documents from each topic) in each category for each graph. For any two graphs in the same category, a document is randomly mapped to documents in the same topic. For the graphs in the different category, the documents are randomly mapped together without considering topics. Thus, the adja-

Table 3: Average NMI of 20-NG.

Method	Comp	Rec	Talk
	$G_1 \sim G_5$	$G_6 \sim G_{10}$	$G_{11} \sim G_{15}$
SymNMF	0.2251	0.2513	0.2473
SC	0.2680	0.2946	0.3333
CTSC	0.5809	**0.6232**	0.8219
PairCRSC	0.3440	0.3928	0.4108
CentCRSC	0.3005	0.3401	0.3704
NoNCLUS	0.3675	0.4150	0.3975
CMNC	**0.5918**	0.6090	**0.9039**

Table 4: Average NMI of different methods on **Digit**.

Method	Feats	Hists	Concat
	$G_1 \sim G_6$	$G_7 \sim G_{10}$	
SymNMF	0.8143	0.2989	0.4722
SC	0.8038	0.3008	0.5282
CTSC	0.7837	0.3148	-
PairCRSC	0.8013	0.3009	0.5458
CentCRSC	0.7663	0.2565	0.5037
CMNC	**0.8591**		

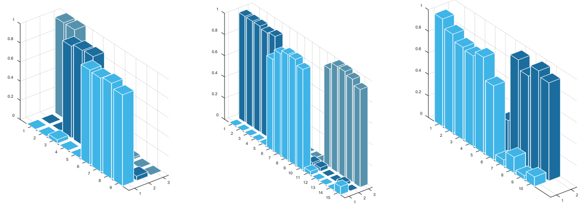
cency tensor is of size 200 × 200 × 15 containing 3 groups of network and 4 document clusters for each group.

Table 3 shows the average NMI over 100 trials for the 20-NG' adjacency tensor. As shown in the table, our methods have a better performance than the other methods in *Comp* and *Talk*. On the other side, without prior network knowledge, our methods reach a comparable level to the state-of-the-art multi-view method, meanwhile group different networks precisely. And it is obvious that the multi-view methods will have a loss when gathering these 15 networks together. We will prove this statement in the next dataset.

Digits

In plenty of scenarios, dataset can be extracted lots of similarity networks by feature engineering. Without prior knowledge to choosing valuable network, we want to separate these networks into two groups: one can provide the desired cluster information while the other is noise. To simulate this case, we have the digits handwritten dataset with six hand-picked Feats: Fourier, profile, Karhunen-Love, pixel, Zernike and morphological. As a helpful feature in image retrieval application, the histogram (Hist) is also extracted from the pixel feature. To construct 5-nearest-neighborhood adjacency matrices, six Feats similarity are measured by Gaussian distance while the Hist one is by four measurements: histogram correlation, Chi-square statistics, histogram intersection and Bhattacharyya distance. These bin-by-bin distances measurements are defined by different theories and have individual characteristics. The adjacency matrices of pixel and histogram correlation are shown in Figure 2e and 2f. Since the noise networks (Hists) are always in a high frequency or high rank, we set $R = 2$ and $L = [10, 20]$ in which 20 is a random number for grouping the noise matrices only.

Table 4 shows the 10 runs results of the network groups: hand-picked features (Feats), histogram similarities (Hists) and their concatenation result (Concat). Compared with SymNMF and SC, the multi-view methods' results of Feats are lower, even though they can learn complementary information from 6 Feats network. On the other hand, from the result of SC, the Hists can still provide some information but very limited, mostly are hindering the performance of multi-view methods. On the contrary, separating noise network in other networks groups and carry out the clustering task on structured networks, CMNC can fully utilize the complementary knowledge without distraction. Thus, CMNC outperforms the other baseline methods. More importantly, our method not only picks out the constructive networks but also provide an initial result for further processing.



(a) **Comp** (b) **20-NG** (c) **Digits**

Figure 3: Matrix $\boldsymbol{C}$ of CMNC in different dataset.

Indicators Analysis

Figure 3 shows the result factor matrix $\boldsymbol{C}$ obtained in different dataset. The clear gap between the correct and the wrong assignment indicating that our method is able to utilize the networks in the same category while filter the graphs of irrelevant categories. Thus, node clustering task can be done individually and the significant improvements are promised.

7 Conclusion

In this paper, we propose a novel tensor decomposition based approach CMNC to solve the multilayer networks clustering with multiple structures. With the tensor representation, CMNC can effectively differentiate irrelevant networks into different groups and captures the underlying clusterings structure from the correlated networks in each group simultaneously. We conduct a thorough discussion and analysis our model theoretically. The effectiveness has been verified by sound experiments.

8 Acknowledgments

The work described in this paper was supported by the National Key R&D Program of China (2018YFB1004804), the National Natural Science Foundation of China (61722214, 11801595), Natural Science Foundation of Guangdong (2018A030310076) and CCF Opening Project of Information System.

References

Cai, D.; Shao, Z.; He, X.; Yan, X.; and Han, J. 2005. Community mining from multi-relational networks. In *PKDD 2005*, volume 3721 of *Lecture Notes in Computer Science*, 445–452. Springer.

Cai, D.; He, X.; Han, J.; and Huang, T. S. 2011. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(8):1548–1560.

Cardillo, A.; Gómez-Gardenes, J.; Zanin, M.; Romance, M.; Papo, D.; Del Pozo, F.; and Boccaletti, S. 2013. Emergence of network features from multiplexity. *Scientific reports* 3:1344.

Chen, C.; Xin, J.; Wang, Y.; Chen, L.; and Ng, M. K. 2018. A semisupervised classification approach for multidomain networks with domain selection. *IEEE Transactions on Neural Networks and Learning Systems* (99):1–15.

Cichocki, A.; Mandic, D. P.; Lathauwer, L. D.; Zhou, G.; Zhao, Q.; Caiafa, C. F.; and Phan, A. H. 2015. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine* 32(2):145–163.

Didier, G.; Brun, C.; and Baudot, A. 2015. Identifying communities from multiplex biological networks. *PeerJ* 3:e1525.

Dong, X.; Frossard, P.; Vandergheynst, P.; and Nefedov, N. 2012. Clustering with multi-layer graphs: A spectral perspective. *IEEE Transactions on Signal Processing* 60(11):5820–5831.

Du, R.; Drake, B.; and Park, H. 2017. Hybrid clustering based on content and connection structure using joint nonnegative matrix factorization. *Journal of Global Optimization*.

Dunlavy, D. M.; Kolda, T. G.; and Kegelmeyer, W. P. 2011. Multilinear algebra for analyzing data with multiple linkages. In *Graph Algorithms in the Language of Linear Algebra*. Society for Industrial and Applied Mathematics. 85–114.

Ficklin, S. P., and Feltus, F. A. 2011. Gene coexpression network alignment and conservation of gene modules between two grass species: maize and rice. *PLANT PHYSIOLOGY* 156(3):1244–1256.

Kivelä, M.; Arenas, A.; Barthelemy, M.; Gleeson, J. P.; Moreno, Y.; and Porter, M. A. 2014. Multilayer networks. *Journal of Complex Networks* 2(3):203–271.

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

Kolda, T. G.; Bader, B. W.; and Kenny, J. P. 2005. Higher-order web link analysis using multilinear algebra. In *ICDM 05*. IEEE.

Kuang, D.; Park, H.; and Ding, C. H. Q. 2012. Symmetric nonnegative matrix factorization for graph clustering. In *SDM 2012*, 106–117. SIAM.

Kumar, A., and III, H. D. 2011. A co-training approach for multi-view spectral clustering. In *ICML 2011*, ICML 2011, 393–400. USA: Omnipress.

Kumar, A.; Rai, P.; and Daume, H. 2011. Co-regularized multi-view spectral clustering. In *NIPS 24*. Curran Associates, Inc. 1413–1421.

Lathauwer, L. D. 2008. Decompositions of a higher-order tensor in block terms—part II: Definitions and uniqueness. *SIAM Journal on Matrix Analysis and Applications* 30(3):1033–1066.

Li, S.; Jiang, Y.; and Zhou, Z. 2014. Partial multi-view clustering. In *AAAI 2014*, 1968–1974. AAAI Press.

Liu, J.; Wang, C.; Gao, J.; and Han, J. 2013a. Multi-view clustering via joint nonnegative matrix factorization. In *SDM 2013*, 252–260. SIAM.

Liu, X.; Ji, S.; Glänzel, W.; and Moor, B. D. 2013b. Multi-view partitioning via tensor methods. *IEEE Transactions on Knowledge and Data Engineering* 25(5):1056–1069.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28(2):129–137.

Newman, M. E. J. 2004. Detecting community structure in networks. *The European Physical Journal B* 38(2):321–330.

Newman, M. E. J. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103(23):8577–8582.

Ni, J.; Tong, H.; Fan, W.; and Zhang, X. 2015. Flexible and robust multi-network clustering. In *KDD 15*. ACM Press.

Paatero, P., and Tapper, U. 1994. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics* 5(2):111–126.

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

Tang, W.; Lu, Z.; and Dhillon, I. S. 2009. Clustering with multiple graphs. In *ICDM 2009*, 1016–1021. IEEE.

Vervliet, N.; Debals, O.; and Lathauwer, L. D. 2016. Tensorlab 3.0 — numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. In *2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE.

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

Wasserman, S., and Faust, K. 1994. *Social Network Analysis*, volume 8. Cambridge university press.

Wright, S., and Nocedal, J. 2006. *Numerical Optimization*. Springer New York.