



## Article

# Community Detection in Multilayer Networks Based on Matrix Factorization and Spectral Embedding Method

Fengqin Tang <sup>1</sup>, Xuejing Zhao <sup>2</sup> and Cuixia Li <sup>3,\*</sup><sup>1</sup> School of Mathematics Sciences, Huaibei Normal University, Huaibei 235000, China<sup>2</sup> School of Mathematics and Statistics, Lanzhou University, Lanzhou 730033, China<sup>3</sup> School of Mathematics and Statistics, Xuzhou University of Technology, Xuzhou 221018, China

\* Correspondence: lcx@xzit.edu.cn

**Abstract:** Community detection remains a challenging research hotspot in network analysis. With the complexity of the network data structures increasing, multilayer networks, in which entities interact through multiple types of connections, prove to be effective in describing complex networks. The layers in a multilayer network may not share a common community structure. In this paper, we propose a joint method based on matrix factorization and spectral embedding to recover the groups not only for the layers but also for nodes. Specifically, the layers are grouped via the matrix factorization method with layer similarity-based regularization in the perspective of a mixture multilayer stochastic block model, and then the node communities within a layer group are revealed by clustering a combination of the spectral embedding derived from the adjacency matrices and the shared approximation matrix. Numerical studies show that the proposed method achieves competitive clustering results as the number of nodes and/or number of layers vary, together with different topologies of network layers. Additionally, we apply the proposed method on two real-world multilayer networks and obtain interesting findings which again highlight the effectiveness of our method.

**Keywords:** community detection; multilayer network; layer similarity; mixture multilayer stochastic block model

MSC: 05C82



**Citation:** Tang, F.; Zhao, X.; Li, C. Community Detection in Multilayer Networks Based on Matrix Factorization and Spectral Embedding Method. *Mathematics* **2023**, *11*, 1573. <https://doi.org/10.3390/math11071573>

Academic Editor: Alfonso Niño

Received: 28 February 2023

Revised: 10 March 2023

Accepted: 22 March 2023

Published: 23 March 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Network analysis has emerged in many areas of research and applications, from sociology and ecology to economics [1,2]. For networks, nodes represent the entities of interest, and connecting edges represent the specific relationship that exists among nodes. As the complexity of real datasets increases, various interactions among the elementary components of these systems have emerged. For example, people may contact each other as workmates in a company and as friends on Facebook; different countries may trade in a variety of products. Traditional monolayer networks, which record only one type of interaction between nodes, are less capable to describe the multiple modes of interactions encountered and to understand the topology of the real datasets precisely. Fortunately, several network models have been in existence as various extensions of monolayer networks. One such example are the multilayer networks in which multiple modes of interactions are considered and each specific interaction forms a layer [3]. In this paper, we focus on the multilayer networks with layers sharing a common set of nodes and without inter-layer edges [4].

Community detection is a fundamental issue when analyzing network data. For monolayer network, community detection aims to cluster nodes into communities, and thus there exist more edges connecting nodes in the same community and fewer edges joining nodes across communities [5]. The community detection problem has significant

applications due to the fact that entities in a community are likely to share common properties; for example, people in a community may come from the same place and/or they may share other similar interests [6]. Thus, revealing the community structure in a network can provide a better understanding of the overall functioning of the network. There exists a substantial volume of research about community detection techniques on monolayers, such as probabilistic methods [7,8] and criteria-based approaches [9,10]. For a comprehensive review of these works we refer to [11].

The community detection methods originally proposed for monolayer networks can be used directly to analyze multilayer networks in two ways. One way is to aggregate the layers into a graph and then apply traditional community detection methods on the aggregated network [12,13]. The other way is to use the traditional community detection methods on each layer independently and the clustering results are then merged [14,15]. Several other researchers have also proposed community detection methods for multilayer networks, including probability models [16–19], spectral methods based on various versions of aggregation adjacency matrices or Laplacian matrices [20,21], matrix factorization methods [22] and criteria based methods such as minimizing a squared objective function or maximizing a modularity function [23–25]. Most of these methods aim to detect communities that are common across layers and ignore the inhomogeneous nature of network layers. However, for many real multilayer networks, different layers may have different community structures, and some of the layers are interdependent while others are not. For instance, in an international trade multilayer network in which products represent layers and countries are nodes, the trade patterns among countries related to processed foods may be different from those related to unprocessed foods; on the other hand, the trade activities among countries related to different processed foods may be similar. These findings suggest that we should jointly find groups for layers and assign community memberships to nodes for each layer group.

A few recent works have focused on addressing the community detection problem in inhomogeneous multilayer networks. For example, Ref. [26] introduce a multilayer SBM which clusters layers into groups, the so-called strata, and assume that layers in each strata share a common community structure; Ref. [27] introduce a common subspace independent-edge multiple random graph model allowing a shared latent structure across layers, being layer-specific in each layer, and propose a joint spectral embedding of adjacency matrices to simultaneously estimate underlying parameters for each layer; Ref. [28] investigate the so called “Mixture MultiLayer Stochastic Block Model” (MMLSBM) and propose a regularized tensor decomposition to reveal the communities for the layers and nodes. Ref. [29] propose an alternating minimization algorithm (ALMA) that simultaneously recovers the layer partitions and node partitions. Although the methods for finding layer groups in the mentioned works are a bit different, given the layer groups, these works apply classical cluster methods either on the aggregated adjacency matrices or an estimated matrix of that layer group to find the node assignments. More related works we refer to are [30–32].

In this work, we are interested in inhomogeneous multilayer networks under the mixture multilayer stochastic block model (MMLSBM) setting. We consider a two-step procedure to detect the layer groups (between-layer clustering) as well as the node communities within a layer group (within-layer clustering). Unlike commonly used methods that describe the multilayer networks using tensor-wise data in the literature, we instead use the vectorization of the model. This technique can reduce the systematic errors due to the improper setting of parameters in the estimation of the tensor approximations. A combination of the Matrix Factorization and Spectral Embedding method (MFSE) is proposed to handle the joint community detection problem. The present paper makes the following contributions. First, under a vectorized perspective, we propose a matrix factorization based method to recover the layer communities. Moreover, since layers that are very structurally similar are more likely to belong to the same group, we integrate layer similarity regularization into the matrix factorization based objective function. Under the MMLSBM setting, each layer can be described as a stochastic block model (SBM) model and two layers

have the same connection probability matrix once they belong to the same layer community. From the least squares perspective, the underlying membership matrix  $C$  of the layers is optimized by minimizing the objective function  $\|\mathbf{A} - CQ\|_F^2 + Pen$ , where  $\mathbf{A}$  is a vectorized version of the adjacency tensor and the  $l$ -th row of matrix  $Q$  can be treated as the potential representation of the  $l$ -th layer community and  $Pen$  is the regularization term. Subsequently, we employ the spectral embedding method to recover the node communities in each layer group. For a given layer group, we simultaneously apply the spectral embedding on the adjacency matrices and on the shared structural matrix derived from  $Q$  in that group. The final node latent representations are a combination of two types of spectral embeddings. Then, classical clustering methods can be carried out on the joint spectral embedding to search the node assignments. In addition, an efficient singular value decomposition is used to give a recommended value of the number of layer groups as well as the number of node communities in each layer group.

The rest of the paper is organized as follows. The framework is illustrated in Section 2. Section 3 gives the theoretical results of the proposed method. Section 4 provides several experimental evaluations to demonstrate the effectiveness of the proposed method and Section 5 summarizes the paper.

## 2. Model Framework

### 2.1. Notations

For a matrix  $A$ , let  $\|A\|_F$  be the Frobenius norm and denote its transpose by  $A^T$ . Let  $A(i; \cdot)$  and  $A(\cdot; j)$  be the  $i$ -th row and  $j$ -th column of a matrix  $A$ , respectively. Denote the Kronecker product of the matrices of matrices  $A$  and  $B$  by  $A \otimes B$ . Some more notations used throughout this paper are collected in Table 1.

**Table 1.** Notation definitions.

| Notations  | Explanations   |
|--|--|
| $[n]$  | $[n] = \{1, 2, \dots, n\}$   |
| $n; L$   | number of nodes, number of network layers  |
| $G^{(l)}; A^{(l)} \in \mathbb{R}^{n \times n}$   | the $l$ -th layer; the corresponding adjacency matrix  |
| $\mathcal{A} \in \mathbb{R}^{n \times n \times L}, \mathcal{A}(\cdot; \cdot; l) = A^{(l)}$ | the adjacency tensor   |
| $M; L_m (\sum_{m=1}^M L_m = L)$  | number of layer groups; number of layers in the $m$ -th layer group  |
| $K_m (m = 1, 2, \dots, M)$   | number of node communities in the $m$ -th layer group  |
| $\mathcal{G}_1, \dots, \mathcal{G}_M$  | the network groups such that $[L] = \bigcup_{m=1}^M \mathcal{G}_m$   |
| $\mathcal{V}_k^{(m)}, m = 1, \dots, M, k = 1, \dots, K_m$                                  | the $k$ -th node community in the $m$ -th layer group  |
| $L_{max}; L_{min}$   | $L_{max} = \max_m L_m; L_{min} = \min_m L_m$   |
| $n_{max}^{(m)}; n_{min}^{(m)}$   | $n_{max}^{(m)} = \max_{k \in \{1, \dots, K_m\}}  \mathcal{V}_k^{(m)} ; n_{min}^{(m)} = \min_{k \in \{1, \dots, K_m\}}  \mathcal{V}_k^{(m)} $ |
| $Z^{(m)} \in \{0, 1\}^{n \times K_m}$  | the membership (clustering) matrix of the $m$ -th layer group  |
| $B^{(m)} \in [0, 1]^{K_m \times K_m}$  | link probability matrix of the $m$ -th layer group   |
| $P^{(m)} = \mathbb{E}(A^{(m)}) = (P_{ij}^{(m)}) \in \mathbb{R}^{n \times n}$               | the expected matrix of the $m$ -th layer   |
| $C \in \{0, 1\}^{L \times M}$  | the membership matrix of layers  |
| $a^{(m)}$  | the vectorized version of matrix $A^{(m)}$ by sequentially stacking its columns  |
| $b^{(m)}, p^{(m)}$   | the vectorized versions of matrices $B^{(m)}, P^{(m)}$ , respectively  |
| $\mathbf{A} \in \mathbb{R}^{L \times n^2}, \mathbf{P} \in \mathbb{R}^{L \times n^2}$       | matrices with rows $a^{(m)}$ and $p^{(m)}$ , respectively  |
| $A(i; \cdot), A(\cdot; j)$   | the $i$ -th row and $j$ -th column of a matrix $A$ , respectively  |
| $A(r; i : j)$  | the $i$ -to- $j$ -th entities in the $r$ -th row of matrix $A$   |
| $L_Q$  | the Laplacian matrix of $Q$  |
| $D_Q$  | a diagonal-matrix with $(i, i)$ -th element $D_Q(i, i) = \sum_j Q(i, j)$   |
| $d_{Q_m}$  | $d_{Q_m} = \min_i D_{Q_m}(i, i)$   |
| $\hat{U}_Q = [\hat{U}_{Q_1}, \dots, \hat{U}_{Q_{K_m}}]; \tilde{U}_Q$                       | eigenvector matrix corresponding to $L_Q$ ; normalized version of $\hat{U}_Q$  |
| $\hat{A}; A^*$   | an estimator of matrix $A$ ; the population version of $A$   |
| $\lambda_k(W)$   | the $k$ -th absolute large eigenvalue of matrix $W$  |
| $\Pi$  | an orthonormal rotation whose rank varies from one line to another   |

## 2.2. Discussion of the Framework

We consider a multilayer network  $\mathcal{G} = \{G^{(1)}, \dots, G^{(L)}\}$  with  $L$  layers on the same set of  $n$  nodes. For the  $l$ -th layer, denote its adjacency matrix by  $A^{(l)} \in \mathbb{R}^{n \times n}$ , where  $A_{ij}^{(l)} = 1$  if the nodes  $i$  and  $j$  are connected and  $A^{(l)}(i, j) = 0$  otherwise, further,  $A_{ij}^{(l)} = A_{ji}^{(l)}$  if the  $l$ -th layer is undirected and  $A_{ij}^{(l)} \neq A_{ji}^{(l)}$  otherwise.

In the MMLSBM setting,  $A_{ij}^{(l)}$  are independent Bernoulli random variables with  $P(A_{ij}^{(l)} = 1) = P_{ij}^{(l)}$  and  $P_{ii}^{(l)} = 0$ . The probability matrices  $P^{(l)}$  give a tensor form  $\mathcal{P} \in \mathbb{R}^{n \times n \times L}$ . In this paper, we assume that the probability connection tensor  $\mathcal{P}$  can be partitioned into  $M$  ( $M < L$ ) groups. In particular, there exists a label function  $c: [L] \rightarrow [M]$  and a corresponding clustering matrix  $C \in \{0, 1\}^{L \times M}$  such that

$$\mathbb{E}[A^{(l)}|Z] = P^{(l)} = Z^{(m)} B^{(m)} (Z^{(m)})^T, m = c(l), 1 \leq l \leq L, 1 \leq m \leq M, \quad (1)$$

where  $B^{(m)} \in [0, 1]^{K_m \times K_m}$  is the connectivity matrix satisfying  $B^{(m)}(i, j) = B^{(m)}(j, i)$ . We aim to recover the network groups  $\mathcal{G}_1, \dots, \mathcal{G}_M$  as well as the within-layer community assignments  $\{\mathcal{V}_k^{(m)}; k = 1, \dots, K_m, m = 1, \dots, M\}$ .

### 2.2.1. Between-Layer Cluster

#### Methods

Adopting the vectorized versions of  $P^{(l)}$  and  $B^{(l)}$ , it follows from [33] that

$$p^{(l)} = (Z^{(c(l))} \otimes Z^{(c(l))}) b^{(c(l))}, l = 1, \dots, L. \quad (2)$$

Denote  $\Theta^{(c(l))} = Z^{(c(l))} \otimes Z^{(c(l))}$  and recall that  $C$  is the membership matrix of the layers, i.e., for  $l = 1, \dots, L; m = 1, \dots, M$ ,  $C(l, m) = 1$  if  $c(l) = m$  and 0 otherwise. We thus rewrite Equation (1) as

$$\mathbf{P} = \begin{pmatrix} p^{(1)} \\ \vdots \\ p^{(L)} \end{pmatrix} = \begin{pmatrix} \Theta^{(c(1))} b^{(c(1))} \\ \vdots \\ \Theta^{(c(L))} b^{(c(L))} \end{pmatrix} = CQ \quad (3)$$

where  $Q = (\Theta^{(1)} b^{(1)}, \dots, \Theta^{(M)} b^{(M)})^T \in \mathbb{R}^{M \times n^2}$  can be considered as the shared structure matrix of  $M$  layer groups.

Similarly, we introduce the matrix  $\mathbf{A} = (a^{(1)}, a^{(2)}, \dots, a^{(L)})^T \in \mathbb{R}^{L \times n^2}$ . Note that  $\mathbf{P}$  can not be observed directly but  $\mathbf{A}$  can be; combining Equations (1)–(3) and from the perspective of the least-squares method, we propose to estimate the potential layer cluster matrix via the following objective function

$$\min_{CC^T = I; Q} \|\mathbf{A} - CQ\|_F^2. \quad (4)$$

In the estimation step, we relax the restrictions on  $C$ , i.e.,  $C(l, m) = 1$  if layer  $l$  belongs to group  $m$ , and impose the non-negativity together with  $CC^T = I$  on  $C$ . For the non-negative feature of the matrices  $\Theta^{(l)}$  and  $b^{(l)}$ , we can solve the optimization problem of Equation (4) by a nonnegative matrix factorization method.

Intuitively, the layers with high pairwise structural similarity tend to be partitioned into a group. Some effort has been devoted to designing various kinds of indices to measure the similarity between layers [34,35]. In this paper, we use the Pearson linear correlation

coefficient  $S(l, m)$  to quantify the degree of correlation between two layers  $l$  and  $m$  [36]. The Pearson correlation coefficient is defined as

$$S(l, m) = \frac{\sum_{ij} (A_{ij}^{(m)} - \overline{A^{(m)}})(A_{ij}^{(l)} - \overline{A^{(l)}})}{\sigma(\overline{A^{(m)}})\sigma(\overline{A^{(l)}})}$$

where  $\overline{A^{(\cdot)}}$  represents the mean degree and  $\sigma(\overline{A^{(\cdot)}})$  is the standard deviation of the layer  $A^{(\cdot)}$ . There is no doubt that  $S(l, m) \in [-1, 1]$ . The closer  $S(l, m)$  is to 1 (−1), the more positively (negatively) correlated the two layers are. Additionally, layers  $l$  and  $m$  are almost linearly independent if  $S(l, m)$  closes to 0.

Above all, a layer similarity based regularization is defined as

$$S(l, m) \| C(l; \cdot) - C(m; \cdot) \|_F^2 \quad (5)$$

where  $S(l, m)$  is the Pearson linear correlation and  $C(l; \cdot)$  represents the  $l$ -th row of matrix  $C$ . A simple calculation leads to the following matrix format of Equation (5)

$$\frac{1}{2} \sum_l \sum_m S(l, m) \| C(l; \cdot) - C(m; \cdot) \|_F^2 = \text{tr}(C^T L_S C) \quad (6)$$

where  $\text{tr}(\cdot)$  is the trace function,  $S = (S(l, m))$  is the layer similarity matrix and  $L_S = D_S - S$  stands for the Laplacian matrix of the layer similarity matrix.  $D_S$  is a diagonal matrix representing the degree matrix of  $S$ .

By integrating the regularization Equation (6) into Equation (4), we now formalize the between-layer cluster problem as a non-negative matrix factorization problem with layer similarity regularization:

$$\min_{CC^T=I, Q} \mathcal{L}(C, Q) = \min_{CC^T=I, Q} \| \mathbf{A} - CQ \|_F^2 + \alpha \text{tr}(C^T L_S C) \quad (7)$$

where  $\alpha$  is a trade-off parameter for the regularization term. In this paper, for  $l \neq m$ , we set the entities  $S(l, m)$  which are less than 0 to 0, and choose the parameter  $\alpha$  in Equation (7) by the cross validation method.

### Updating Rules

Note that  $\mathcal{L}(C, Q)$  is not convex with respect to  $C$  and  $Q$  together; it is thus difficult to find the global minimum of the loss function. To this end, an iterative algorithm is adopted to pursue the solution of Equation (7). In addition, we introduce Lagrange multipliers  $\Gamma \in \mathbb{R}^{L \times M}$  and  $\Xi \in \mathbb{R}^{M \times n^2}$  for the non-negative constraints on  $C$  and  $Q$ , respectively. Then the objective function Equation (7) is rewritten as

$$\min_{CC^T=I, Q} \mathcal{L}(C, Q) = \min_{CC^T=I, Q} \| \mathbf{A} - CQ \|_F^2 + \alpha \text{tr}(C^T L_S C) + \text{tr}(\Gamma C^T) + \text{tr}(\Xi Q^T). \quad (8)$$

The partial derivatives of Equation (8) with respect to  $C$  and  $Q$  are

$$\frac{\partial \mathcal{L}(C, Q)}{\partial C} = 2CQQ^T - 2\mathbf{A}Q^T + 2\alpha L_S C + \Gamma. \quad (9)$$

and

$$\frac{\partial \mathcal{L}(C, Q)}{\partial Q} = 2C^T CQ - 2C^T \mathbf{A} + \Xi. \quad (10)$$

A local minimum of Equation (8) is achieved by respectively setting

$$\frac{\partial \mathcal{L}(C, Q)}{\partial C} = 0 \quad (11)$$

and

$$\frac{\partial \mathcal{L}(C, Q)}{\partial Q} = 0. \quad (12)$$

After that, for  $l \in \{1, 2, \dots, L\}$ ,  $m \in \{1, 2, \dots, M\}$  and  $i \in \{1, 2, \dots, n^2\}$ , we multiply by  $C(l, m)$  and  $Q(m, i)$  on both sides of Equations (11) and (12), respectively. Since the KKT conditions imply that  $C(l, m)\Gamma(l, m) = 0$  and  $Q(m, i)\Xi(m, i) = 0$ , we obtain the optimal solutions of  $C(l, m)$  and  $Q(m, i)$  as follows:

$$\hat{C}(l, m) \leftarrow C(l, m) \frac{(\mathbf{A}Q^T)(l, m)}{(CQQ^T + \alpha L_S C)(l, m)} \quad (13)$$

and

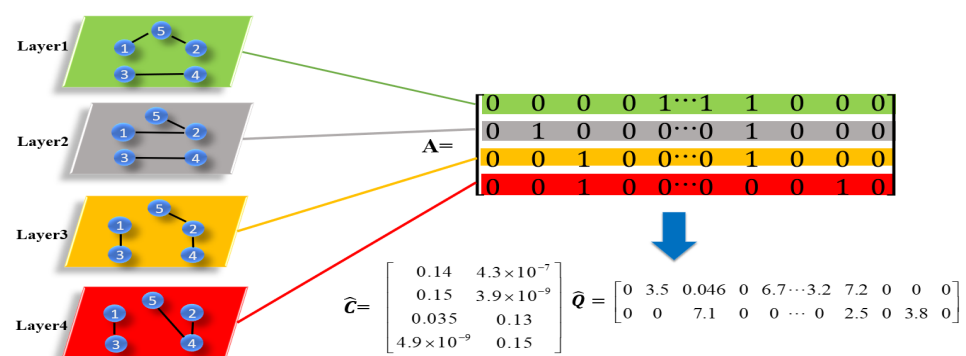
$$\hat{Q}(m, i) \leftarrow Q(m, i) \frac{(C^T \mathbf{A})(m, i)}{(C^T C Q)(m, i)}. \quad (14)$$

Now, we can recover the communities of network layers by applying K-means algorithm on the rows of  $\hat{C}$ . For each  $m$ , the estimated  $\hat{Q}(m; \cdot)$  represents the shared structural information of layers in the  $m$ -th layer group; we transform  $\hat{Q}$  into  $M$  separate matrices. Specifically, for  $m = 1, \dots, M$ , we consider matrices  $\hat{Q}_m \in \mathbb{R}^{n \times n}$  given by

$$\hat{Q}_m(i; \cdot) = \hat{Q}(m; ((i-1)n+1) : in). \quad (15)$$

We stress that  $\hat{Q}_m(i; \cdot)$  represents the shared behavior of node  $i$  in the  $m$ -th layer group.

A toy example is further used to describe the procedure of Equation (7). As shown in Figure 1, the simple multilayer network consists of four layers and five nodes. Each row of the matrix  $\mathbf{A} \in \mathbb{R}^{4 \times 25}$  represents the vectorized version of the adjacency matrix related to the layer. Applying the updating rules in Equations (13) and (14), we obtain the estimated layer membership matrix  $C \in \mathbb{R}^{4 \times 2}$  and the estimated shared structure information matrix  $Q \in \mathbb{R}^{2 \times 25}$ . After that, we use K-means on  $\hat{C}$  to obtain the layer groups. Then, by Equation (15), we construct two  $5 \times 5$  matrices  $\hat{Q}_1$  and  $\hat{Q}_2$  based on two rows of  $\hat{Q}$ , respectively. Each constructed matrix is treated as the potential representation of the related layer group.



**Figure 1.** A toy example describes how to estimate  $C$  and  $Q$  in the objective function Equation (7).



### 2.2.2. Within-Layer Community Detection

Given the layer membership matrix  $\hat{C}$  and the shared structure matrix  $\hat{Q}$ , the issue that we focus on is how to use the structural information within a layer to detect node assignments. Refs. [28,29] applied a classical clustering method either on the aggregated adjacency matrices or an estimated embedding matrix associated to that layer group. To better describe the consistent topology structural across layers within a layer group, we propose to use an aggregated spectral embedding derived from both the observed adjacency matrices in that layer group and of the shared  $\hat{Q}_m$  on the task of clustering nodes within a layer group.

Once the group layers are fixed, we average the adjacency matrices in the  $m$ -th group as  $\bar{A}_m = \frac{\sum_{l:c(l)=m} A^{(l)}}{L_m}$ . Then, a standard spectral embedding method is applied on  $\bar{A}_m$ . We list the detailed steps below.

(Sa): Calculate the graph Laplacian  $L_{\bar{A}_m} = D_{\bar{A}_m}^{1/2} \bar{A}_m D_{\bar{A}_m}^{1/2}$ .

(Sb): Find orthonormal eigenvectors  $\hat{U}_{\bar{A}_m 1}, \hat{U}_{\bar{A}_m 2}, \dots, \hat{U}_{\bar{A}_m K_m}$  corresponding to the  $K_m$  eigenvalues that are largest in absolute value of  $L_{\bar{A}_m}$  and put them in a matrix  $\hat{U}_{\bar{A}_m} = [\hat{U}_{\bar{A}_m 1}, \hat{U}_{\bar{A}_m 2}, \dots, \hat{U}_{\bar{A}_m K_m}] \in \mathbb{R}^{n \times K_m}$ .

(Sc): Form the normalized version  $\tilde{U}_{\bar{A}_m}$  by normalizing each row of  $\hat{U}_{\bar{A}_m}$  to have unit length. Then,  $\tilde{U}_{\bar{A}_m}$  is considered to be the spectral embedding derived from  $\bar{A}_m$ .

Following the steps (Sa)–(Sc) of the spectral embedding, we are coming to find the embedding representations of nodes from  $\hat{Q}_m$ . Denote the graph Laplacian by  $L_{\hat{Q}_m} = D_{\hat{Q}_m}^{1/2} \hat{Q}_m D_{\hat{Q}_m}^{1/2}$  and the corresponding normalized embedding matrix by  $\tilde{U}_{\hat{Q}_m}$ .

Based on the above, for the  $m$ -th layer group, we add the node representations obtained in  $\tilde{U}_{\bar{A}_m}$  with the  $\tilde{U}_{\hat{Q}_m}$  to obtain extended node representations:

$$\hat{Y}_m = \frac{\tilde{U}_{\bar{A}_m} + \tilde{U}_{\hat{Q}_m}}{2}. \quad (16)$$

The within-layer community detection is then carried out by applying a K-means algorithm on  $\hat{Y}_m$ . Up to now, we formalize a two-step community detection method for multilayer networks based on Matrix Factorization and joint Spectral Embedding (MFSE) which is outlined as Algorithm 1.

**Remark 1 (Initializations of C and Q).** It is known that random initializations may lead the algorithms into non-informative local minimals. The Pearson correlation matrix  $S$  illustrates the structural similarity among layers and thus it can provide a glancing estimation of the membership of the layers. Here, we apply K-means on the layer similarity matrix  $S$  and take the centers returned by the K-means algorithm as the initialization of  $C$ . In our proposed objective function, each row of  $Q$  stands for the potential representations of nodes in a specific layer group, following the way of initializing  $C$ ; after carrying out K-means on  $A$ , the initialization of  $Q$  is given by the centers returned by the K-means algorithm.

**Remark 2 (Pre-determination of M and  $K_m(m = 1, \dots, M)$ ).** Much work has been carried out for determining the number of communities in the community detection problem ([37,38]), either by maximizing specific objective function or searching all candidates. Among these, the eigengap heuristic method proves to be effective in applications ([39,40]). In this paper, we select the number of layer groups  $M$  and number of node communities  $K_m$  of the  $m$ -th layer group by the eigengap heuristic method. For  $M$ , we compute the eigendecomposition of the layer similarity matrix  $S$  and then rank the eigenvalues  $\lambda_1(S) > \lambda_2(S) > \dots$ . In practice, there is usually a significant drop between  $\lambda_M(S)$  and  $\lambda_{M+1}(S)$  and we set the number of layer groups as  $M$ .

Once the layer groups are fixed, we first obtain the averaged adjacency matrix  $\bar{A}_m$  in the  $m$ -th layer group and then apply eigen-decomposition on  $\bar{A}_m$ . After that,  $K_m$  is chosen as the number of

communities if the eigenvalue  $\lambda_{K_m}(\bar{A}_m)$  is relatively larger than  $\lambda_{K_m+1}(\bar{A}_m)$ , i.e., the gap between  $\lambda_{K_m}(\bar{A}_m)$  and  $\lambda_{K_m+1}(\bar{A}_m)$  is much greater than those of other adjacent eigenvalues.

---

**Algorithm 1** MFSE Algorithm
 

---

**Input:** The adjacency matrices  $A^{(l)}, l = 1, 2, \dots, L$ ; number of layer groups  $M$ ; number of the node groups  $\{K_m\}_{m=1}^M$ ; tuning parameter  $\alpha$ .

**Output:** The network groups  $\{\mathcal{G}_1, \dots, \mathcal{G}_M\}$  and the within-layer node communities  $\{\mathcal{V}_k^{(m)}; m = 1, \dots, M, k = 1, \dots, K_m\}$ .

**Algorithm:**

1. Initialize  $C$  and  $Q$ ;
  2. Iterate until convergence.
    - (a2). Update  $\hat{C}$  according to Equation (13);
    - (b2). Update  $\hat{Q}$  according to Equation (14);
  3. Applying K-means on the rows of  $\hat{C}$  to get the layer groups  $\mathcal{G}_1, \dots, \mathcal{G}_M$ .
  4. For each layer group  $\mathcal{G}_m$ .
    - (a4). Applying the spectral embedding method, i.e., steps (Sa)–(Sc) on  $\bar{A}_m$  to obtain the latent node representation matrix  $\tilde{U}_{\bar{A}_m}$ ;
    - (b4). Similar to (a4), estimate the node representation matrix  $\tilde{U}_{\hat{Q}_m}$  based on  $\hat{Q}_m$ ;
    - (c4). The layer groups  $\{\mathcal{V}_k^{(m)}\}$  are obtained by applying K-means on the rows of the final node embedding  $\hat{Y}_m$  in Equation (16).
- 

### 3. Theoretical Guarantees

Some additional assumptions are required to analyze the convergence issues. We collect them as follows:

- (A1)  $d_{Q_m} > 3 \log \frac{8n}{\epsilon}$  for  $\epsilon > 0$ ;
- (A2)  $24d_{Q_m}^{1/2} \lambda_{K_m} \geq (3 \log \frac{8n}{\epsilon})^{1/2}$  with  $\lambda_{K_m}$  representing the  $K_m$ -th largest eigenvalue of  $L_{Q_m} L_{Q_m}$ ;
- (A3) cluster of layers and local communities balance condition: there exist positive constants  $\underline{\gamma} < \bar{\gamma}$  such that

$$\underline{\gamma}M < \frac{L}{L_{\max}} \leq \frac{L}{L_{\min}} < \bar{\gamma}M, \underline{\gamma}K_m < \frac{n}{n_{\max}^{(m)}} \leq \frac{n}{n_{\min}^{(m)}} < \bar{\gamma}K_m.$$

Note that Assumption (A1) restricts the sparsity of the network, Assumption (A2) requires that the  $K_m$ -th eigenvalue of  $L_{Q_m} L_{Q_m}$  is not too small and Assumption (A3) is the so-called balanced network condition and is very common in the problem of community detection ([41]).

We need to obtain the estimators  $\hat{C}$  and  $\hat{Q}$  of  $C^* \in \mathbb{R}^{L \times M}$  and  $Q^* \in \mathbb{R}^{M \times n^2}$  based on the observed  $\{A^{(l)}\}$ . First of all, we give a theoretical result on the consistency of the estimators from the objective function  $\mathcal{L}(C, Q)$ . We denote the estimators in the iteration by  $\{\hat{\Delta}^{[t]} = (\hat{C}^{[t]}, \hat{Q}^{[t]})\}_{t=1}^{\infty}$ .

**Theorem 1.**  $\{\hat{\Delta}^{[t]}\}_{t=1}^{\infty}$  converges to a critical point of  $\mathcal{L}(C, Q)$ .

The proof of Theorem 1 is given in Appendix A.



We denote by  $\pi(m)$  the permutation function of  $[m]$ . Given the true partition of the network layers  $[L] = \bigcup_{m=1}^M \mathcal{G}_m$  and the estimated partition  $[L] = \bigcup_{m=1}^M \hat{\mathcal{G}}_m$ , the misclassification error rate of between-layer clustering can be defined as

$$\text{Err}_{BL} = \min_{\tau \in \pi(m)} \sum_{m=1}^M |\mathcal{G}_m \setminus \hat{\mathcal{G}}_{\tau(m)}| / L. \quad (17)$$

**Theorem 2.** Suppose that Assumption (A3) holds, there exists a sufficiently small function  $\varepsilon(n, L)$  depending on  $n$  and  $L$  such that the between-layer clustering error satisfies  $\text{Err}_{BL} \leq \frac{\zeta_\varepsilon \varepsilon(n, L) \max_m L_m}{4L}$ , where  $\zeta_\varepsilon$  is a positive constant depending on  $\varepsilon$ .

The proof of Theorem 2 is given in Appendix B.

For the  $m$ -th layer group, the cluster centroids sought by K-means algorithm are defined as  $\mathcal{O} = \{O_m \in \mathbb{R}^{n \times d} : O_m \text{ has } K_m \text{ unique rows}\}$ , satisfying  $\hat{O}_m = \arg \min_{\hat{X}_m \in \mathcal{O}} \|\hat{Y}_m - O_m\|_F$ . Let

$$\Gamma_m = \left\{ i : \|\hat{O}_m(i; \cdot) - Y_m^*(i; \cdot) \Pi\|_2 > \|\hat{O}_m(j; \cdot) - Y_m^*(j; \cdot) \Pi\|_2 \right\} \quad (18)$$

be the set of misclustered nodes, where  $Y_m^*(i; \cdot)$  represents the parameter centroid corresponding to the  $i$ -th node while  $\hat{O}_m(i; \cdot)$  is the estimated centroid, and  $\Pi$  is an orthonormal rotation.

**Theorem 3.** Assume that the smallest eigenvalue of  $E\bar{A}_m$  is larger than  $\alpha_n$  and Assumptions (A1)–(A2) hold. There exist constants  $C(\varepsilon) > 0, C_0 > 0, 0 < \xi \leq 1$  and for any  $\varepsilon > 0$ , with probability at least  $1 - \varepsilon$ ; the proportion of misclustered nodes in the  $m$ -th layer group satisfies  $\frac{|\Gamma_m|}{n} \leq \frac{\xi}{2} \left( \frac{144C_0K_m \log(8n/\varepsilon)}{n\lambda_{K_m}^2} + \frac{C(\varepsilon)K_m \log n}{n\alpha_n^2} \right)$ .

The proof of Theorem 3 is given in Appendix C.

## 4. Experiments

In this section, we assess the performance of the proposed MFSE method through simulated and real-world multi-layer networks. For synthetic multilayer networks, we compare our method with the TWIST method ([28]) and the ALMA method ([29]). Specifically, TWIST focuses on the MMSBM model and applies Tucker decomposition on the observed adjacency tensor to obtain its low-rank approximations. TWIST finds both the global and the local memberships of nodes, together with the memberships of layers. ALMA is an alternating minimization algorithm which aims at revealing the layer partition as well as the node partitions. We remark that both MFSE and ALMA aim at recovering the community assignment of layers together with the nodes. To make the comparison fair, all three competitive methods are only dedicated to detecting the layer partition and the node partition which are also named as “between-layer clustering” and “within-layer clustering”, respectively. For both synthetic multilayer networks and real-world networks, the between-layer clustering error is assessed by  $\text{Err}_{BL}$  (Equation (17)) and the proportion of misclassified nodes within layers (the within-layer clustering error) is evaluated by  $\frac{\sum_{m=1}^M |\Gamma_m|}{Mn}$  (Equation (18)).

### 4.1. Simulation Study

In this part, we test our method on synthetic multilayer networks. We carry out three simulations to evaluate the performance of the proposed method by varying the number of layers  $L$ , the number of nodes  $n$  and network sparsity parameter  $\gamma$ , respectively. The

clustering results of all methods in three simulations averaged over 500 replications. Below, we elaborate on each of the simulation scenarios.

The purpose of the first simulation is to examine the ability of the proposed method to discover both layer and node clusters as the number of nodes  $n$  varies. Imagine a MMLSBM with  $L = 60$  layers,  $M = 3$  layer groups and  $K_1 = K_2 = K = 2$  node communities for each layer group. The underlying community assignments of layers and nodes are generated by the multinomial distributions with equal class probabilities  $1/M$  for the layers, and  $1/K$  for the nodes within a layer group. For completeness, we take into account both assortative and disassortative SBM models ([42]) in the tests. A SBM model is assortative if  $B(i, j) < B(i, i)$  and disassortative if  $B(i, j) > B(i, i)$  for  $i \neq j$ , with  $B(i, j)$  representing the connection probability ([41]). We set the connection matrices for three layer groups as

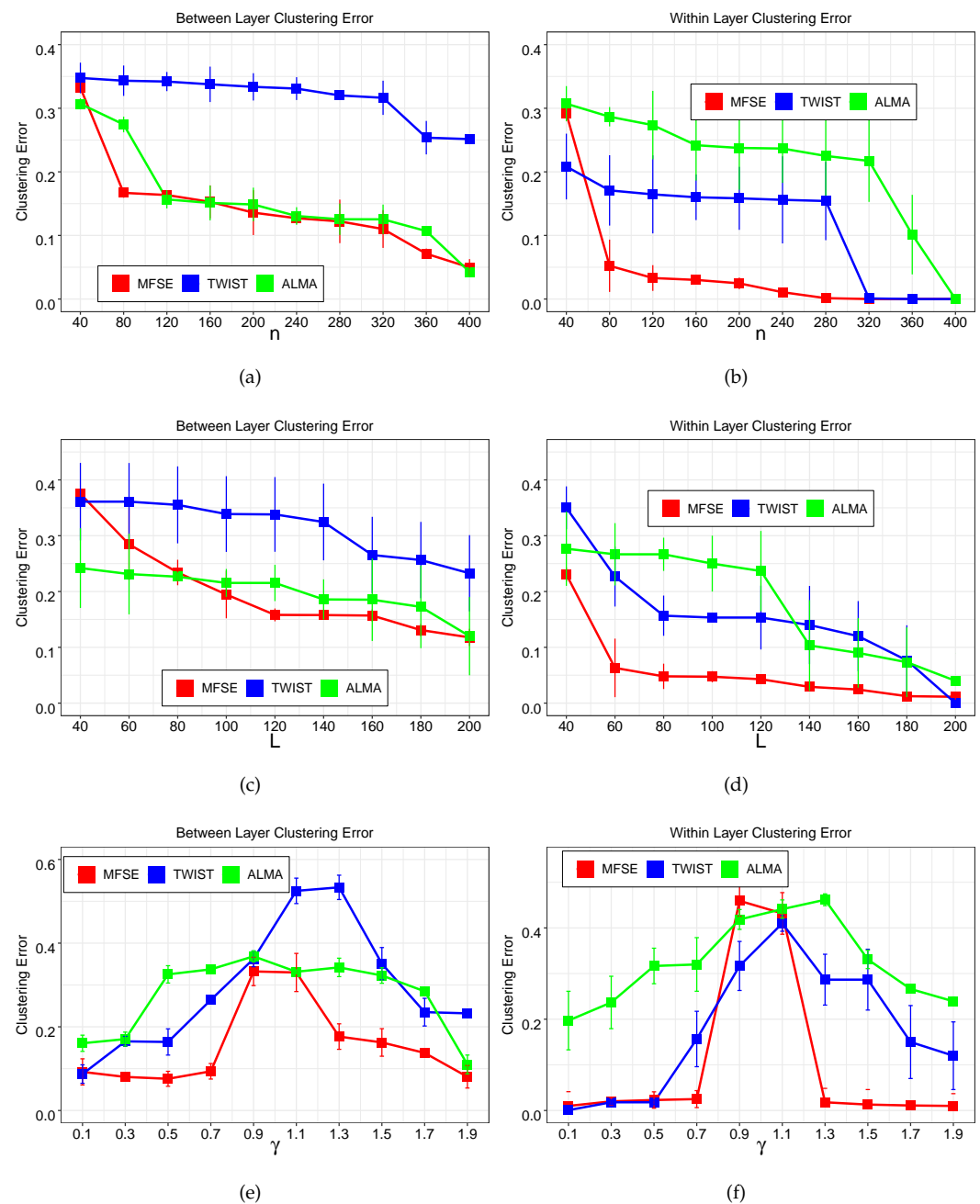
$$\mathbf{B}_m = \begin{pmatrix} p_m & \gamma_m p_m & \gamma_m p_m \\ \gamma_m p_m & p_m & \gamma_m p_m \\ \gamma_m p_m & \gamma_m p_m & p_m \end{pmatrix}, m = 1, 2, 3 \quad (19)$$

where  $p = (p_1, p_2, p_3) = (0.4, 0.4, 0.5)$  and  $\gamma = (\gamma_1, \gamma_2, \gamma_3) = (1.2, 0.8, 0.6)$ . Notice that values of  $\gamma_m$  govern the strength of the assortative (disassortative) feature of the network. Roughly speaking, the network is assortative if  $\gamma_m < 1$  and disassortative if  $\gamma_m > 1$ , and the community structure becomes obscure as  $\gamma_m$  closes to 1. We increase the number of nodes  $n$  from 40 to 400 and record the between-layer clustering error results and within-layer clustering error results of three methods in Figure 2a,b. As expected, both types of errors rates for all three methods decrease substantially as  $n$  becomes large, implying the effect of number of nodes on clustering errors. As  $n \geq 80$ , the plots demonstrate that MFSE always achieves superior or comparative performances in both between-layer and within-layer clustering.

In the second simulation, we evaluate the performances of three methods as the number of layers varies. The generated MMLSBM is similar to that in the first simulation except  $n = 100$  is fixed. We set the number of layers  $L$  to increase from 40 to 200. The clustering errors are reported in Figure 2c,d. For all methods, both types of errors decrease as  $L$  increases, which is consistent with the theoretical results of these methods. Specifically, for between-layer clustering, MFSE outperforms TWIST and ALMA once  $L \geq 80$ , and MFSE continues to obtain superior results for the within-layer clustering.

The last simulation studies the impact of parameter  $\gamma$  which controls the ratio of the connection probabilities of the inter-groups of nodes versus intra-groups of nodes. The settings are the same as previous simulation scenarios except  $n = 100$  and  $L = 120$  are fixed. We set  $\gamma = \{\gamma_1, \gamma_1, \gamma_1\}$ , and the performances of all methods on two types of clustering with  $\gamma_1$  increasing from 0.1 to 1.9 are investigated. The results of this comparison are presented in Figure 2e,f. Notice that the smaller the  $\gamma_1$ , the more pronounced the within-group and between-group distinction becomes. Additionally, the larger  $\gamma_1 (> 1)$  indicates the stronger disassortative feature of the network. Further, the within-layer clustering becomes harder when  $\gamma_1$  closes to 1, which also makes the between-layer clustering difficult. The performances of the three methods in Figure 2e,f agree with those findings.

In summary, for between-layer clustering, the proposed method MFSE achieves better results in all parameter settings and ALMA almost gives the second best results. For within-layer clustering, MFSE still gives superior or comparable performances in different settings. We claim that TWIST works notably even in the cases where the between-layer errors are higher than those of ALAM. This could be due to the fact that TWIST recovers the node partitions by clustering the adjacency matrices in that layer group, while ALAM clusters the low-rank approximation of the adjacency tensor in that layer group which may leave out some direct structure information. We use both direct adjacency matrices and the low-rank approximations to reveal the node memberships within a layer group, which makes our method more effective in various model settings.



**Figure 2.** (a,b) Between-layer clustering results and within-layer clustering results returned by three methods as the number of nodes  $n$  increases; (c,d) Between-layer clustering results and within-layer clustering results returned by three methods as the number of layers  $L$  increases; (e,f) Between-layer clustering results and within-layer clustering results returned by three methods as  $\gamma$  varies. Each data point shows the average result and the error bar determines the range of one standard deviation from the average.

#### 4.2. Real-World Network Data

In this section, we demonstrate the merits of MFSE in community detection on two real-world multilayer networks.

##### 4.2.1. FAO data

The data are collected from the Worldwide Food Trading Networks data that are available at <https://www.fao.org/faostat/en/#data/TM>. The data was collected since 1986

and the last update was given in 23 December 2022. In our analysis, the data was accessed on 1 January 2010, before 31 December 2010. The dataset contains export/import trading among 245 countries/regions for more than 300 products. In the multilayer network, layers represent different types of commodities and nodes represent countries. Two countries have a link if there exist import/export relationships of a product between them. We convert the original directed and weighted networks to undirected and unweighted ones by ignoring the directions and weights. We first remove the layers with links less than 1135 (the third quartile) and retain the layers with average degree greater than 10 (the second quartile). Finally, we choose the countries that are active in trading at least 90% of the selected products, reducing a multilayer network consisting of 46 layers and 100 nodes.

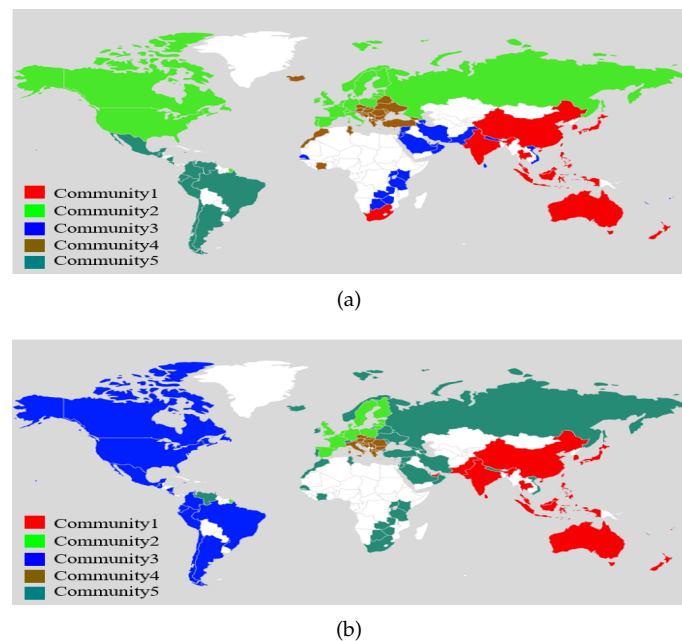
Subsequently, we use Algorithm 1 to recover the partitions of layers (products) and nodes (countries) of the multilayer network. By applying the eigengap heuristic method, we choose  $M = 2$  and record the between-layer clustering results in Table 2.

**Table 2.** Between-layer clustering results obtained by MFSE.

| Layer Groups | Products  |
|--------------|---|
| Group 1      | Cheese_whole_cow_milk, Bread, Chocolate_products_nes, Fatty_acids, Anise_badian_fennel_coriander, Beer_of_barley, Chillies_and_peppers_dry, Beverages_distilled_alcoholic, Coffee_extracts, Coffee_roasted, Fruit_dried_nes, Juice_fruit_nes, Pepper_(piper_spp.), Sugar_nes, Tea_mate_extracts, Vegetables_frozen  |
| Group 2      | Beverages_non_alcoholic, Macaroni, Food_prep_nes, Cigarettes, Beans_dry, Flour_wheat, Crude_materials, Food_preparations_flour_malt_extract, Food_wastes, Fruit_prepared_nes, Cereals_breakfast, Coffee_green, Cocoa_powder_and_cake, Pastry, Pet_food, Tea, Nuts_prepared_(exc_groundnuts), Mixes_and_doughs, Oil_vegetable_origin_nes, Oil_essential_nes, Rice_milled, Sugar_refined, Sugar_confectionery, Wine, Spices_nes, Vegetables_dehydrated, Vegetables_preserved_nes, Waters_ice_etc, Vegetables_in_vinegar, Tobacco_unmanufactured |

From Table 2, we see that the first layer group consists mostly of solid and liquid beverage products as well as processed food products with a certain shelf life, and the second group contains mostly cereals and raw materials for food products, together with processed food products with a long shelf life.

Applying the eigengap heuristic method again, we choose  $K_1 = K_2 = 5$ , which is consistent with the number of continents (Africa, Americas, Asia, Europe, and Oceania) and can be considered as a natural partition of countries. Figure 3 shows the node communities recovered by MFSE. We have the following comments: (1) for both layer groups, there exist close trading activities among some developed and developing countries in Asia and Oceania, for example, China, Japan, Australia, New Zealand, etc, see Community 1 in Figure 3a,b, respectively; (2) for the products in group 1, since North America and Europe are not far apart by ocean transportation, countries in these two regions have more continual demands for trade than the countries in the Americas which have a longer shipping distance, please refer to Communities 2 and 3 in Figure 3a; (3) for the products in group 2, the considerable shelf life of some products allows for close trade among countries with long marine transportation distance, such as among countries in North America and South America (Community 3 in Figure 3b); (4) the countries in Southeast Europe (Community 4 in Figure 3a,b, respectively) have their independent trade patterns regardless of the types of products, probably due to the Free Trade Agreement which was carried out in 2006.



**Figure 3.** (a) Node communities found in layer group 1 by MSFE; (b) Node communities found in layer group 2 by MSFE.

#### 4.2.2. AU-CS Network

The network is constructed from a dataset that describes five online and offline relationships including Facebook, leisure, work, coauthor and lunch among 61 employees of a university research department ([43]). The employees are represented by nodes and one specific relationship is represented by one layer, together with links standing for an existing relation between nodes. Thus, a five-layer multilayer network with 620 edges is formed. As a pre-processing step, we computed the eigenvalues of the graph Laplacian based on the layer-similarity matrix and chose  $M = 2$  by eigengap heuristic. After that, by using the eigengap heuristic method again, we chose  $K_1 = K_2 = 3$  as the numbers of node communities in two layer groups, respectively. We apply Algorithm 1 on the data and list the between-layer clustering result in Table 3. The intuition of these layer partitions is that the employees have a natural co-occurrence of work and lunch interactions as colleagues; for example, they may work together and have lunch together in the canteen. Simultaneously, it is natural to group together the online/offline relationships based on personal/research interests.

Then, we investigate the within-layer clustering results. Unlike the FAO data, we may not explain intuitively the node partitions and have no idea about the groundtruth communities. For completeness, we use the density metric to evaluate the quality of the clustering method. The density at the  $l$ -th network layer is defined as follows:

$$density = \sum_{k=1}^{K_l} \frac{|\{(v_i^{(l)}, v_j^{(l)}) | v_i^{(l)}, v_j^{(l)} \in C_k^{(l)}, (v_i^{(l)}, v_j^{(l)}) \in E^{(l)}\}|}{|E^{(l)}|}$$

where, for the  $l$ -th layer,  $v_i^{(l)}$ ,  $C_k^{(l)}$  and  $E^{(l)}$  represent the  $i$ -th node, the  $k$ -th node community and the edge set, respectively. Additionally, to make the results comparable, we introduce a standard Spectral Clustering based method (SPC-Mean) in the literature which uses

the Mean adjacency matrix  $\bar{A} = \frac{\sum_{l=1}^L A^{(l)}}{L}$  to recover the node communities. We record the densities of each layer returned by the proposed MFSE method and SPC-Mean method in Table 4. We further plot the adjacency matrix of each layer returned by MFSE (Figure 4) where nodes are arranged according to the community labels. Table 4 shows that MFSE

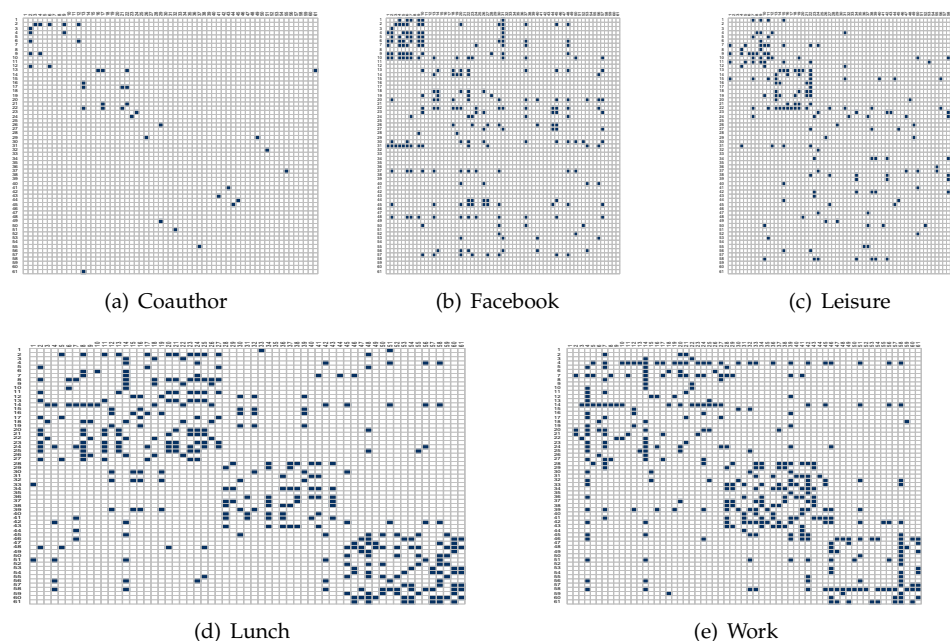
gives highly competitive densities for all layers on the AU-CS network and Figure 4 implies that the connection patterns of the first three layers are significantly different from those of the remaining two layers.

**Table 3.** Between-layer clustering results obtained by MFSE on AU-CS network.

| Layer Groups | Layers                      |
|--------------|-----------------------------|
| Group 1      | Coauthor, Leisure, Facebook |
| Group 2      | Lunch, Work                 |

**Table 4.** Densities obtained by MFSE and SPC-Mean on each layer of AU-CS network.

| Layers   | MFSE | SPC-Mean |
|----------|------|----------|
| Coauthor | 0.95 | 0.76     |
| Facebook | 0.76 | 0.70     |
| Leisure  | 0.83 | 0.76     |
| Lunch    | 0.72 | 0.62     |
| Work     | 0.77 | 0.71     |



**Figure 4.** (a–e) Adjacency matrices of five layers on AU-CS network with nodes arranged according to the community labels obtained by MFSE.

## 5. Conclusions

In this paper, we propose a matrix factorization and spectral embedding based community detection method for multilayer networks. The proposed method recovers the layer groups by clustering the low-rank approximate of a vectorized version matrix derived by the adjacency matrices. After that, the node communities within a layer group can be recovered by both the observed adjacency matrices and a shared structural matrix in that layer group. We give the theoretical guarantees of the method and the simulation results confirm the fact that the proposed method performs efficiently under different model settings. Notice that the layers of networks may have various degrees of sparsity in many real applications; it is thus of interest to extend our method to more sparse scenarios in future studies. Another direction worth exploring is where one can generalize the matrix



factorization-based representation incorporating node attributes to extract communities in multilayer networks.

**Author Contributions:** Conceptualization, F.T.; methodology, F.T.; software, X.Z.; validation, X.Z.; formal analysis, X.Z.; investigation, F.T.; resources, C.L.; data curation, C.L.; writing—original draft preparation, C.L.; writing—review and editing, C.L.; visualization, C.L.; supervision, C.L.; project administration, F.T.; funding acquisition, F.T. and X.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China(12201235, 11971214), Natural Science Foundation of Anhui Province (2108085QA14), the Excellent Young Talents Fund Program of Higher Education Institutions of Anhui Province (gxyqZD2022044) and Qinglan Project of Jiangsu Province.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank the editor, associate editor and referees.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

|        |   |
|--------|---|
| MFSE   | Matrix factorization and spectral embedding method      |
| MMLSBM | Mixture multilayer stochastic block model               |
| TWIST  | Tucker decomposition with integrated SVD transformation |
| ALMA   | Alternating minimization algorithm                      |

## Appendix A

The proof of Theorem 1 is based on the following lemmas and the Kurdyka–Lojasiewicz property.

**Lemma A1.** *There exists a constant  $\iota > 0$  such that for  $t = 1, 2, \dots$ ,  $\|\hat{\Delta}^{[t]}\|_F^2 \leq \iota$ .*

**Proof.** Since  $\mathcal{L}(C, Q) \geq 0$  and the sequence of  $\{\Delta^{[t]}\}_{t=1}^\infty$  is monotone decreasing, the result is obtained.  $\square$

**Lemma A2.** *For each  $t = 1, 2, \dots$ , it holds that  $\iota_1 \|\hat{\Delta}^{[t+1]} - \hat{\Delta}^{[t]}\|_F^2 \leq (\mathcal{L}(\hat{\Delta}^{[t]}) - \mathcal{L}(\hat{\Delta}^{[t+1]}))$ , where  $\iota_1 = 2 \min\{\alpha^2(\lambda_1(L_S))^2, 1\}$  and  $\lambda_1(L_S)$  is the largest eigenvalue of  $L_S$ .*

**Proof.** Denote  $\hat{C} = \text{vec}(\hat{C}^{[t]} - \hat{C}^{[t+1]})$ . For each  $t = 1, 2, \dots$ , we have

$$\begin{aligned} & \mathcal{L}(\hat{C}^{[t]}, \hat{Q}^{[t]}) - \mathcal{L}(\hat{C}^{[t+1]}, \hat{Q}^{[t]}) \\ &= \left\langle \frac{\partial \mathcal{L}(\hat{C}^{[t+1]}, \hat{Q}^{[t]})}{\partial \hat{C}}, \hat{C} \right\rangle + 2\hat{C}^T \left( (\hat{Q}^{[t]})^T \hat{Q}^{[t]} \otimes I_{M^2 \times M^2} + \alpha L_S^T \otimes I_{n^2 \times n^2} \right) \hat{C}_F^2 \\ & \geq 2\|\hat{C}^{[t]} - \hat{C}^{[t+1]}\|_F^2 \alpha^2 (\lambda_1(L_S))^2 \end{aligned} \quad (\text{A1})$$

where  $\otimes$  is the Kronecker product of two matrices.

Similarly, we have

$$\begin{aligned} & \mathcal{L}(\hat{C}^{[t]}, \hat{Q}^{[t]}) - \mathcal{L}(\hat{C}^{[t]}, \hat{Q}^{[t+1]}) \\ &= \left\langle \frac{\partial \mathcal{L}(\hat{C}^{[t]}, \hat{Q}^{[t+1]})}{\partial \hat{Q}}, \hat{Q}^{[t]} - \hat{Q}^{[t+1]} \right\rangle + 2(\hat{Q}^{[t]} - \hat{Q}^{[t+1]})^T (\hat{Q}^{[t]} - \hat{Q}^{[t+1]})_F^2 \\ & \geq 2\|\hat{Q}^{[t]} - \hat{Q}^{[t+1]}\|_F^2 \end{aligned} \quad (\text{A2})$$

Combining Equations (A1) and (A2) yields that  $\iota_1 \|\hat{\Delta}^{[t+1]} - \hat{\Delta}^{[t]}\|_F^2 \leq (\mathcal{L}(\hat{\Delta}^{[t]}) - \mathcal{L}(\hat{\Delta}^{[t+1]}))$ .  
□

**Lemma A3.** For any  $t = 1, 2, \dots$  and  $\forall l^{[t+1]} \in \partial \mathcal{L}(\hat{\Delta}^{[t+1]})$ , it holds that  $\|l^{[t+1]}\|_F \leq \iota_2 \|\hat{\Delta}^{[t]} - \hat{\Delta}^{[t+1]}\|_F$ , where  $\iota_2 = 2(\iota^2 + \|A\|_F)$ .

**Proof.** It follows from Equation (11) that

$$\frac{\partial \mathcal{L}(\hat{\Delta}^{[t+1]})}{\partial \hat{C}} = 2\hat{C}^{[t+1]} \hat{Q}^{[t+1]} (\hat{Q}^{[t+1]})^T - 2\mathbf{A}(\hat{Q}^{[t+1]})^T + 2\alpha L_S \hat{C}^{[t+1]} + \Gamma. \quad (\text{A3})$$

On the other hand,

$$\begin{aligned} \frac{\partial \mathcal{L}(\hat{C}^{[t+1]}, \hat{Q}^{[t]})}{\partial \hat{C}} &= 2\hat{C}^{[t+1]} \hat{Q}^{[t]} (\hat{Q}^{[t]})^T - 2\mathbf{A}(\hat{Q}^{[t]})^T + 2\alpha L_S \hat{C}^{[t+1]} + \Gamma \\ &= 0. \end{aligned} \quad (\text{A4})$$

Combining Equations (A3) and (A4), we can get

$$\begin{aligned} &\left\| \frac{\partial \mathcal{L}(\hat{\Delta}^{[t+1]})}{\partial \hat{C}} \right\|_F \\ &\leq 2\|\hat{C}^{[t+1]} \hat{Q}^{[t+1]} (\hat{Q}^{[t+1]})^T - \hat{C}^{[t+1]} \hat{Q}^{[t]} (\hat{Q}^{[t]})^T\|_F + 2\|\mathbf{A}(\hat{Q}^{[t+1]})^T - \mathbf{A}(\hat{Q}^{[t]})^T\|_F \\ &\leq 2\iota^2 \|\hat{Q}^{[t+1]} - \hat{Q}^{[t]}\|_F + 2\|\mathbf{A}\|_F \|\hat{Q}^{[t+1]} - \hat{Q}^{[t]}\|_F \\ &\leq 2(\iota^2 + \|\mathbf{A}\|_F) \|\hat{Q}^{[t+1]} - \hat{Q}^{[t]}\|_F \end{aligned} \quad (\text{A5})$$

Similarly, the computation of Equation (10) implies that

$$\frac{\partial \mathcal{L}(\hat{\Delta}^{[t+1]})}{\partial \hat{Q}} = 2(\hat{C}^{[t+1]})^T \hat{C}^{[t+1]} \hat{Q}^{[t+1]} - 2(\hat{C}^{[t+1]})^T \mathbf{A} + \Xi. \quad (\text{A6})$$

and

$$\frac{\partial \mathcal{L}(\hat{C}^{[t]}, \hat{Q}^{[t+1]})}{\partial \hat{Q}} = 2(\hat{C}^{[t]})^T \hat{C}^{[t]} \hat{Q}^{[t+1]} - 2(\hat{C}^{[t+1]})^T \mathbf{A} + \Xi = 0. \quad (\text{A7})$$

Plugging Equation (A6) into Equation (A5) yields that

$$\begin{aligned} &\left\| \frac{\partial \mathcal{L}(\hat{\Delta}^{[t+1]})}{\partial \hat{Q}} \right\|_F \\ &\leq 2\|\hat{C}^{[t+1]} \hat{C}^{[t+1]} (\hat{Q}^{[t+1]})^T - \hat{C}^{[t]} \hat{C}^{[t]} (\hat{Q}^{[t+1]})^T\|_F + 2\|\mathbf{A} \hat{C}^{[t+1]} - \mathbf{A} \hat{C}^{[t]}\|_F \\ &\leq 2\gamma^2 \|\hat{C}^{[t+1]} - \hat{C}^{[t]}\|_F + 2\|\mathbf{A}\|_F \|\hat{C}^{[t+1]} - \hat{C}^{[t]}\|_F \\ &\leq 2(\gamma^2 + \|\mathbf{A}\|_F) \|\hat{C}^{[t+1]} - \hat{C}^{[t]}\|_F. \end{aligned} \quad (\text{A8})$$

Together with Equation (A5) and Equation (A7), we complete the proof. □

Let  $x, \mathcal{X}$  be a point and a point set, respectively. Define  $\text{dist}(x, \mathcal{X}) = \min_{y \in \mathcal{X}} \text{dist}(x, y)$  as the shortest distance between  $x$  and  $\mathcal{X}$ . Denote  $\Psi_v$  the class of all concave and continuous functions  $\varrho: [0, v] \rightarrow (0, +\infty)$  satisfying: (1)  $\varrho(0) = 0$ ; (2)  $\varrho$  is continuous differentiable on  $(0, v)$  and (3) for all  $\dot{v} \in (0, v)$ ,  $\varrho'(\dot{v}) > 0$ . Now, we are presenting the Kurdyka–Lojasiewicz (KL) property.

**Definition A1.** (Kurdyka–Lojasiewicz (KL) property [44]) Let  $\kappa: \mathbb{R}^d \rightarrow (-\infty, +\infty]$  be a proper lower semicontinuous function. For  $\dot{x} \in \text{dom} \partial \kappa = \{x \in \mathbb{R}^d : \partial \kappa(x) \neq \emptyset\}$  if there exists an

$v \in (0, +\infty]$ , a neighborhood  $\mathbb{X}$  of  $\dot{x}$  and a function  $\varrho \in \Psi_v$  such that for all  $x \in \mathbb{X} \cap \{\kappa(\dot{x}) < \kappa(x) < \kappa(\dot{x}) + v\}$ , it holds that

$$\varrho'(\kappa(x) - \kappa(\dot{x})) \text{dist}(0, \partial\kappa(x)) > 1,$$

then  $\kappa$  satisfies the KL property at  $\dot{x}$ . Further, if  $\kappa$  satisfies the KL property at each point in  $\text{dom}\partial\kappa$ , we then call  $\kappa$  a KL function.

**Proof of Theorem 1.** The proof of Theorem 1 is based on the sufficient conditions provided by Lemmas A1–A3 and the KL property in Definition A1. An analogous analysis of convergence about the integrative matrix factorization problem was completed in [45]. The proof of Theorem 1 can be conducted according to Theorem 1 in [45]. We omit the details here.  $\square$

## Appendix B

The proof of Theorem 2 requires the following lemma simplified from [32], which provides an error bound for any  $(1 + \varepsilon)$ -approximate K-means solution.

**Lemma A4.** Let  $\Psi \in \mathbb{R}^{n \times d}$  with  $K$  distinct rows satisfying  $\min_{i \neq j} \|\Psi(i; \cdot) - \Psi(j; \cdot)\|_F^2 \geq \gamma$  with  $\gamma$  a positive constant. Let  $\hat{\Psi} \in \mathbb{R}^{n \times d}$  and  $\hat{Y}$  be an  $(1 + \varepsilon)$  solution to K-means problem with input  $\hat{\Psi}$ ; then, the number of errors in  $\hat{Y}$  is no larger than  $\zeta_\varepsilon \|\Psi - \hat{\Psi}\|_F^2 \gamma^{-2}$  for some constant  $\zeta_\varepsilon$  depending only on  $\varepsilon$ .

**Proof of Theorem 2.** It follows from Theorem 1 that there exist a sufficiently small function  $\varepsilon(n, L)$  depending on  $n$  and  $L$  such that  $\|\hat{C} - C^*\|_F^2 \leq \varepsilon(n, L)$ . Note that  $C^* \in \mathbb{R}^{L \times M}$  with  $\min \|C^*(i; \cdot) - C^*(j; \cdot)\| \geq \frac{2}{\sqrt{L_{\max}}}$ . Combining this fact with Lemma A4, we conclude that the misclassification rate is not larger than  $\frac{\zeta_\varepsilon L_{\max}}{4L} \|\hat{C} - C^*\|_F^2 \leq \frac{\zeta_\varepsilon \varepsilon(n, L) L_{\max}}{4L}$ .  $\square$

## Appendix C

**Proof of Theorem 3.** Denote  $\max\{\max_i \|O_m(i; \cdot)\|, \max_i \|Y_m^*(i; \cdot)\|\} = \xi (\xi \leq 1)$ . Define  $\tilde{\Gamma}_m = \{i : \|O_m(i; \cdot) - Y_m^*(i; \cdot)\|_2 \geq \frac{1}{\sqrt{\xi}}\}$ . It is clear that

$$\begin{aligned} |\Gamma_m| &\leq |\tilde{\Gamma}_m| = \sum_{i \in \tilde{\Gamma}_m} 1 \leq \sqrt{\xi} \sum_{i \in \tilde{\Gamma}_m} \|\hat{O}_m(i; \cdot) - Y_m^*(i; \cdot)\|_2 \leq \xi \|\hat{O}_m - Y_m^* \Pi\|_F^2 \\ &\leq \xi \left( \|\hat{O}_m - \hat{Y}_m \Pi\|_F^2 + \|\hat{Y}_m - Y_m^* \Pi\|_F^2 \right) \leq 2\xi \|\hat{Y}_m - Y_m^* \Pi\|_F^2 \\ &\leq \frac{\xi}{2} \left( \|\tilde{U}_{\tilde{A}_m} - Y_m^* \Pi\|_F^2 + \|\tilde{U}_{\tilde{Q}_m} - Y_m^* \Pi\|_F^2 \right) = \frac{\xi}{2} (I_1 + I_2) \end{aligned}$$

We first deal with  $I_1$ . For some constant  $C(\varepsilon) > 0$ , it immediately follows from Theorem 3.1 of [46] that

$$I_1 \leq \frac{C(\varepsilon) K_m \log n}{\alpha_n^2}. \quad (\text{A9})$$

Under Assumptions (A1) and (A2), Theorem 3 of [47] confirms that

$$I_2 \leq \frac{144 C_0 K_m \log(8n/\varepsilon)}{\lambda_{K_m}^2} \quad (\text{A10})$$

A combination of Equations (A9) and (A10) lead to the desired result.  $\square$

## References

- Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [\[CrossRef\]](#)
- Javed, M.A.; Younis, M.S.; Latif, S.; Qadir, J.; Baig, A. Community detection in networks: A multidisciplinary review. *J. Netw. Comput. Appl.* **2018**, *108*, 87–111. [\[CrossRef\]](#)
- Kim, J.; Lee, J.G. Community detection in multi-layer graphs: A survey. *ACM Sigmod Rec.* **2015**, *44*, 37–48. [\[CrossRef\]](#)
- Kivelä, M.; Arenas, A.; Barthélemy, M.; Gleeson, J.P.; Moreno, Y.; Porter, M.A. Multilayer networks. *J. Complex Netw.* **2014**, *2*, 203–271. [\[CrossRef\]](#)
- Hric, D.; Darst, R.K.; Fortunato, S. Community detection in networks: Structural communities versus ground truth. *Phys. Rev. E* **2014**, *90*, 062805. [\[CrossRef\]](#) [\[PubMed\]](#)
- Traud, A.L.; Kelsic, E.D.; Mucha, P.J.; Porter, M.A. Comparing community structure to characteristics in online collegiate social networks. *SIAM Rev.* **2011**, *53*, 526–543.
- Mariadassou, M.; Robin, S.; Vacher, C. Uncovering latent structure in valued graphs: A variational approach. *Ann. Appl. Stat.* **2010**, *4*, 715–742. [\[CrossRef\]](#)
- Amini, A.A.; Chen, A.; Bickel, P.J.; Levina, E. Pseudo-likelihood methods for community detection in large sparse networks. *Ann. Stat.* **2013**, *41*, 2097–2122. [\[CrossRef\]](#)
- Newman, M.E.J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [\[CrossRef\]](#)
- Karrer, B.; Newman, M.E. Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **2011**, *83*, 016107. [\[CrossRef\]](#)
- Fortunato, S.; Darko, H. Community detection in networks: A user guide. *Phys. Rep.* **2016**, *659*, 1–44. [\[CrossRef\]](#)
- Taylor, D.; Caceres, R.S.; Mucha, P.J. Super-resolution community detection for layer-aggregated multilayer networks. *Phys. Rev. X* **2017**, *7*, 031056. [\[CrossRef\]](#) [\[PubMed\]](#)
- Taylor, D.; Shai, S.; Stanley, N.; Mucha, P.J. Enhanced detectability of community structure in multilayer networks through layer aggregation. *Phys. Rev. Lett.* **2016**, *116*, 228301. [\[CrossRef\]](#) [\[PubMed\]](#)
- Dong, X.; Frossard, P.; Vandergheynst, P.; Nefedov, N. Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. *IEEE Trans. Signal. Proces.* **2013**, *62*, 905–918. [\[CrossRef\]](#)
- Tang, L.; Wang, X.; Liu, H. Community detection via heterogeneous interaction analysis. *Data Min. Knowl. Discov.* **2012**, *25*, 1–33. [\[CrossRef\]](#)
- Xu, K.S.; Hero, A.O. Dynamic stochastic blockmodels for time-evolving social networks. *IEEE J. Sel. Top. Signal Process.* **2014**, *8*, 552–562. [\[CrossRef\]](#)
- Han, Q.; Kevin, X.; Edoardo, A. Consistent estimation of dynamic and multi-layer block models. In Proceedings of the International Conference on Machine Learning, Miami, FL, USA, 9–11 December 2015; pp. 1511–1520.
- Tang, R.; Tang, M.; Vogelstein, J.T.; Priebe, C.E. Robust estimation from multiple graphs under gross error contamination. *arXiv* **2017**, arXiv:1707.03487.
- De Bacco, C.; Power, E.A.; Larremore, D.B.; Moore, C. Community detection, link prediction, and layer interdependence in multilayer networks. *Phys. Rev. E* **2017**, *95*, 042317. [\[CrossRef\]](#) [\[PubMed\]](#)
- Dong, X.; Frossard, P.V.; Ergheynst, P.; Nefedov, N. Clustering with multi-layer graphs: A spectral perspective. *IEEE. Trans. Signal. Proces.* **2012**, *60*, 5820–5831. [\[CrossRef\]](#)
- Bhattacharyya, S.; Chatterjee, S. Spectral clustering for multiple sparse networks: I. *arXiv* **2018**, arXiv:1805.10594.
- Tang, W.; Lu, Z.; Dhillon, I.S. Clustering with multiple graphs. In Proceedings of the Ninth IEEE International Conference on Data Mining, Miami, FL, USA, 6–9 December 2009; pp. 1016–1021.
- Lei, J.; Chen, K.; Lynch, B. Consistent community detection in multi-layer network data. *Biometrika* **2020**, *107*, 61–73. [\[CrossRef\]](#)
- Mucha, P.J.; Richardson, T.; Macon, K.; Porter, M.A.; Onnela, J.P. Community structure in time-dependent, multiscale, and multiplex networks. *Science* **2010**, *328*, 876–878. [\[CrossRef\]](#)
- Bazzi, M.; Porter, M.A.; Williams, S.; McDonald, M.; Fenn, D.J.; Howison, S.D. Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Model. Simul.* **2016**, *14*, 1–41. [\[CrossRef\]](#)
- Stanley, N.; Shai, S.; Taylor, D.; Mucha, P.J. Clustering network layers with the strata multilayer stochastic block model. *IEEE. Trans. Netw. Sci. Eng.* **2016**, *3*, 95–105. [\[CrossRef\]](#)
- Arroyo, J.; Athreya, A.; Cape, J.; Chen, G.; Priebe, C.E.; Vogelstein, J.T. Inference for multiple heterogeneous networks with a common invariant subspace. *J. Mach. Learn. Res.* **2021**, *22*, 6303–6351.
- Jing, B.Y.; Li, T.; Lyu, Z.; Xia, D. Community detection on mixture multilayer networks via regularized tensor decomposition. *Ann. Stat.* **2021**, *49*, 3181–3205. [\[CrossRef\]](#)
- Fan, X.; Pensky, M.; Yu, F.; Zhang, T. ALMA: Alternating Minimization Algorithm for Clustering Mixture Multilayer Network. *J. Mach. Learn. Res.* **2022**, *23*, 6303–6351.
- Le, C.M.; Levin, K.; Levina, E. Estimating a network from multiple noisy realizations. *Electron. J. Stat* **2018**, *12*, 4697–4740. [\[CrossRef\]](#)
- Chen, S.; Liu, S.; Ma, Z. Global and individualized community detection in inhomogeneous multilayer networks. *Ann. Stat.* **2022**, *50*, 2664–2693. [\[CrossRef\]](#)
- Lei, J.; Lin, K.Z. Bias-adjusted spectral clustering in multi-layer stochastic block models. *J. Am. Stat. Assoc.* **2022**, 1–13. [\[CrossRef\]](#)
- Gupta, A.K.; Nagar, D.K. *Matrix Variate Distributions*; Chapman Hall/CRC Press: Boca Raton, FL, USA, 2000.

34. De Domenico, M.; Granell, C.; Porter, M.A.; Arenas, A. The physics of spreading processes in multilayer networks. *Nat. Phys.* **2016**, *12*, 901–906. [[CrossRef](#)]
35. De Domenico, M.; Biamonte, J. Spectral entropies as information-theoretic tools for complex network comparison. *Phys. Rev. X* **2016**, *6*, 041062. [[CrossRef](#)]
36. Nicosia, V.; Latora, V. Measuring and modeling correlations in multiplex networks. *Phys. Rev. X* **2015**, *92*, 032805. [[CrossRef](#)] [[PubMed](#)]
37. Yang, L.; Cao, X.; Jin, D.; Wang, X.; Meng, D. A unified semi-supervised community detection framework using latent space graph regularization. *IEEE Trans. Cybern.* **2015**, *45*, 2585–2598. [[CrossRef](#)] [[PubMed](#)]
38. Ma, X.; Gao, L.; Yong, X.; Fu, L. Semi-supervised clustering algorithm for community structure detection in complex networks. *Phys. A* **2010**, *389*, 187–197. [[CrossRef](#)]
39. Luxburg, V. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [[CrossRef](#)]
40. Lu, H.; Sang, X.; Zhao, Q.; Lu, J. Community detection algorithm based on nonnegative matrix factorization and pairwise constraints. *Phys. A* **2020**, *545*, 123491. [[CrossRef](#)]
41. Amini, A.A.; Levina, E. On semidefinite relaxations for the block model. *Ann. Stat.* **2018**, *46*, 149–179. [[CrossRef](#)]
42. Tang, F.; Wang, C.; Su, J.; Wang, Y. Spectral clustering-based community detection using graph distance and node attributes. *Computation. Stat.* **2020**, *35*, 69–94. [[CrossRef](#)]
43. Rossi, L.; Magnani, M. Towards effective visual analytics on multiplex and multilayer networks. *Chaos. Soliton. Fract.* **2015**, *72*, 68–76. [[CrossRef](#)]
44. Attouch, H.; Bolte, J.; Svaiter, B.F. Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized gauss-seidel methods. *Math. Program.* **2013**, *137*, 91–129. [[CrossRef](#)]
45. Liu, J.; Wang, J.; Liu, B. Community detection of multi-Layer attributed networks via penalized alternating factorization. *Mathematics* **2020**, *8*, 239. [[CrossRef](#)]
46. Lei, J.; Rinaldo, A. Consistency of spectral clustering in stochastic block models. *Ann. Stat.* **2015**, *43*, 215–237. [[CrossRef](#)]
47. Binkiewicz, N.; Vogelstein, J.T.; Rohe, K. Covariate-assisted spectral clustering. *Biometrika* **2017**, *104*, 361–377. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.