symmetry

MDPI

Article

A Unified Spectral Clustering Approach for Detecting Community Structure in Multilayer Networks

Esraa Al-sharoa [1,*] and Selin Aviyente [2]

[1] Electrical Engineering Department, Jordan University of Science and Technology, Irbid 22110, Jordan
[2] Electrical and Computer Engineering Department, Michigan State University, East Lansing, MI 48824, USA; aviyente@egr.msu.edu
* Correspondence: emalsharoa@just.edu.jo

Abstract: Networks offer a compact representation of complex systems such as social, communication, and biological systems. Traditional network models are often inadequate to capture the diverse nature of contemporary networks, which may exhibit temporal variation and multiple types of interactions between entities. Multilayer networks (MLNs) provide a more comprehensive representation by allowing interactions between nodes to be represented by different types of links, each reflecting a distinct type of interaction. Community detection reveals meaningful structure and provides a better understanding of the overall functioning of networks. Current approaches to multilayer community detection are either limited to community detection over the aggregated network or are extensions of single-layer community detection methods with simplifying assumptions such as a common community structure across layers. Moreover, most of the existing methods are limited to multiplex networks with no inter-layer edges. In this paper, we introduce a spectral-clustering-based community detection method for two-layer MLNs. The problem of detecting the community structure is formulated as an optimization problem where the normalized cut for each layer is minimized simultaneously with the normalized cut for the bipartite network along with regularization terms that ensure the consistency of the within- and across-layer community structures. The proposed method is evaluated on both synthetic and real networks and compared to state-of-the-art methods. MLNs. The problem of detecting the community structure is formulated as an optimization problem where the normalized cut for each layer is minimized simultaneously with the normalized cut for the bipartite network along with regularization terms that ensure the consistency of the intra- and inter-layer community structures. The proposed method is evaluated on both synthetic and real networks and compared to state-of-the-art methods.

Keywords: multilayernetworks; bipartite graph; spectral clustering; projection distance

check for updates

Citation: Al-sharoa, E.; Aviyente, S. A Unified Spectral Clustering Approach for Detecting Community Structure in Multilayer Networks. *Symmetry* **2023**, *15*, 1368. https://doi.org/10.3390/sym15071368

Academic Editor: Jun Wu

Received: 11 May 2023
Revised: 21 June 2023
Accepted: 3 July 2023
Published: 5 July 2023

1. Introduction

Networks provide a compact representation of the internal structure of complex systems consisting of agents that interact with each other. Some example application areas include social sciences, engineering systems, and biological systems [1,2]. A core task in network analysis is community detection, which identifies the partition of the node set such that within-community connections are denser than between-community connections.

While different methods have been proposed to detect the community structure of simple (single-layer) graphs, in many contemporary applications, a pair of nodes may interact through multiple types of links yielding multilayer networks. In MLNs, each type of link represents a unique type of interaction. These links can be separated into different layers, enabling the same group of nodes to be connected in multiple ways [3]. The layers in a multilayer network can represent various attributes or features of a complex system. For instance, they can be temporal snapshots of the same network at different time intervals, or they can correspond to different types of connections in social networks, e.g., friendship, collaboration, or family relationships, different types of units in military tactical networks, e.g., infantry, vehicles, or airborne units [4], or transportation networks, where nodes representing different locations can be linked through various modes of transportation, including roads, railways, and air routes. Multilayer networks can further be categorized based on the homogeneity of the nodes and complexity of topological structure as (i) multiplex networks that exhibit homogeneity in terms of the entities they comprise, with each layer consisting of the same set of entities of the same type where the inter-layer edges are implicit and not shown; (ii) heterogeneous multilayer networks by the possibility of having different sets and types of entities for each layer and the relationships between entities across layers are explicitly represented by inter-layer edges.

Current approaches to multilayer community detection are either limited to community detection over the aggregated network or are extensions of single-layer community detection methods with simplifying assumptions such as a common community structure across layers. Moreover, they are mostly limited to multiplex networks with no inter-layer edges.

In this paper, we extend the notion of spectral clustering from single-layer networks to two-layer networks with inter-layer edges. In particular, we model the two-layer network as the union of two single-layer networks and a bipartite network that is represented through its symmetric rows' and columns' adjacency matrices. Next, we express the cost functions corresponding to minimizing the normalized cut for each of the layers as well as the inter-layer adjacency matrix in its relaxed form similar to spectral clustering. In order to ensure the consistency of the communities across layers, we regularize the resulting cost function by a projection distance metric that quantifies the consistency of the low-rank embeddings of the networks across layers. The resulting optimization problem is solved through an alternating maximization scheme.

2. Related Work

Community detection methods for multilayer networks can be broadly categorized into three classes: flattening methods, aggregation methods, and direct methods. Flattening methods convert the multilayer network into a single-layer network by collapsing the layers and then apply traditional community detection algorithms. This approach ignores the information present in the multiple layers, which can lead to loss of important features and inaccuracies in the community structure [5]. Aggregation methods detect the community structure for each layer separately and then merge the results into a single structure. This method requires a merging strategy to combine the community structures from each layer, which can be challenging and subjective. Additionally, it may fail to capture the inter-layer dependencies and correlations between the layers [6]. Direct methods work directly on the multilayer network and optimize community-quality assessment criteria such as modularity or normalized cut to identify the community structure. This approach accounts for the interactions between the layers and can reveal the inter-layer dependencies and correlations [7–9].

Some examples of the direct method include multilayer label propagation, random-walk-based methods, non-negative matrix factorization, modularity, and spectral-clustering-based methods. Label propagation algorithms (LPAs) propagate node attributes based on their neighbors' behavior and exhibit linear complexity. Inspired by the traditional LPA, the authors of [10] presented a redefinition of the neighborhood in multilayer networks and proposed a multilayer LPA. Although this approach is efficient and can handle weighted and directed networks, the resulting partition is highly dependent on the threshold parameter and the density of the network dataset. Moreover, this method is only suitable for multiplex networks. Kuncheva et al. [9] proposed locally adaptive random transitions (LARTs), which are designed to detect communities that are shared by some or all layers in multiplex networks. More recently, matrix and tensor factorization methods have been proposed for multilayer community detection. Among these, non-negative matrix factorization (NMF)-based methods that extract low-dimensional feature representations for each layer, where collective factorization is then used to fuse them into a common representation [11]. In [12], a semi-supervised joint non-negative matrix factorization (S2-jNMF) algorithm is proposed for community detection in multiplex networks, aiming to detect a common structure across layers. However, all of these methods are restricted to multiplex networks, where inter-layer edges are only allowed between each node and its corresponding replicas across different layers.

More recently, community detection methods that consider fully connected MLNs, MLNs with inter-relations, have been proposed [13]. The authors of [14,15] propose to extend the modularity function and its solution to account for MLNs with inter-layer relations. Similarly, [16] propose a normalized cut extension to MLNs by creating a block Laplacian matrix, where each block corresponds to a specific layer. The community structure is then obtained through standard spectral clustering on this block Laplacian matrix. However, the selection of the parameter β is crucial to ensuring the community structure consistency across the layers in this method. Another commonly employed technique that incorporates the concept of network dynamics, specifically diffusion, is Infomap [17]. This method optimizes the map equation, which leverages the information–theoretic relationship between reducing network dimensionality and detecting network communities. However, in the case of noisy networks, the efficiency of the diffusion process, i.e., information propagation, may be compromised, leading to suboptimal clustering performance [18].

3. Background

3.1. Graph Theory

Single-layer network: A single-layer network or a simple graph models the interactions between entities in network science. A single-layer network can be defined as $\mathcal{G}_S = (V_S, E_S, \mathbf{A})$, where V_S is the set of nodes, $E_S \subseteq (V_S \times V_S)$ is the set of edges, and $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the symmetric adjacency matrix, with $n = |V|$ being the number of nodes.

Bipartite network: A bipartite network or graph is a graph that can be partitioned into two sets of vertices where all edges connect vertices from one set to vertices in the other set. Formally, a bipartite graph is defined as $\mathcal{G}_B = (V_B, E_B, \mathbf{B})$, where $V_B = V_1 \cup V_2$, $E_B \subseteq (V_1 \times V_2)$ are the disjoint sets of vertices and the set of edges that connects a vertex in V_1 to a vertex in V_2. $\mathbf{B} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ denotes the symmetric adjacency matrix of the bipartite network, where $n_1 = |V_1|$ and $n_2 = |V_2|$ refer to the size of the two disjoint sets. The adjacency matrix of the bipartite graph can be defined as follows:

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{A}^{12} \\ \mathbf{A}^{21} & \mathbf{0} \end{bmatrix}, \tag{1}$$

where $\mathbf{A}^{12} \in \mathbb{R}^{n_1 \times n_2}$ describes the relationships between V_1 and V_2.

Two-layer network: A two-layer network is a type of multilayer network that consists of two layers or graphs, where each layer represents a different type of relationship or interaction between nodes. A two-layer network, $\mathcal{G}_{2M}$, can be formally defined as the set of two single-layer graphs, $\mathcal{G}_1$ and $\mathcal{G}_2$, and a bipartite graph, $\mathcal{G}_{12}$, such that $\mathcal{G}_{2M} = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_{12}\}$ with $\mathcal{G}_1 = (V_1, E_1, \mathbf{A}^1)$, $\mathcal{G}_2 = (V_2, E_2, \mathbf{A}^2)$, and $\mathcal{G}_{12} = (V_1, V_2, E_{12}, \mathbf{A}^{12})$. $\mathcal{G}_1$ and $\mathcal{G}_2$ are known as within- or intra-layer graphs, whereas $\mathcal{G}_{12}$ refers to the across- or inter-layer graph.

Supra-adjacency matrix: A supra-adjacency matrix is a symmetric matrix that represents both the intra- and inter-layer connections in a multilayer network. A supra-adjacency matrix of a two-layer MLN, $\mathbf{A}_{2M} \in \mathbb{R}^{N \times N}$, where $N = n_1 + n_2$ can be constructed from the intra- and inter-layer adjacency matrices as follows:

$$\mathbf{A}_{2M} = \begin{bmatrix} \mathbf{A}^1 & \mathbf{A}^{12} \\ \mathbf{A}^{21} & \mathbf{A}^2 \end{bmatrix}, \tag{2}$$

where $\mathbf{A}^1 \in \mathbb{R}^{n_1 \times n_1}$, $\mathbf{A}^2 \in \mathbb{R}^{n_2 \times n_2}$, and $\mathbf{A}^{12} \in \mathbb{R}^{n_1 \times n_2}$.

3.2. Graph Cut Problem and Spectral Clustering

Graph minimum cut (mincut) is a problem in graph theory that involves partitioning a graph into multiple partitions or disjoint sets of nodes such that the number of edges between these sets is minimized. The mincut problem is NP-hard. However, there are efficient algorithms to approximate the mincut, such as spectral clustering. Spectral clustering relies on the spectral properties (eigenvalues and eigenvectors) of the symmetric graph Laplacian matrix or normalized adjacency matrix. In particular, spectral clustering uses these eigenvalues and eigenvectors to embed the nodes of the graph into a lower-dimensional space. The nodes can be then clustered using standard k-means in this lower-dimensional space.

Given a single-layer graph with a symmetric adjacency matrix, $\mathbf{A} \in \mathbb{R}^{n \times n}$, spectral clustering solves the following trace maximization problem [19,20]:

$$\max_{\mathbf{U} \in \mathbb{R}^{n \times k}} tr\left(\mathbf{U}^{\top} \mathbf{A}_N \mathbf{U}\right), \qquad s.t \;\; \mathbf{U}^{\top}\mathbf{U} = \mathbf{I}_k, \tag{3}$$

where "tr" and "$\top$" refer to the trace and transpose operators, respectively. $\mathbf{A}_N = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ is the normalized version of the adjacency matrix, where $\mathbf{D}$ is the degree matrix with $D_{ii} = \sum_j A_{ij}$. Spectral clustering uses the spectrum (eigenvalues) of the normalized adjacency matrix to partition the nodes into clusters. In particular, the eigenvectors corresponding to the largest k eigenvalues are used to embed the nodes in a low-dimensional space, where the matrix $\mathbf{U}$ is constructed by arranging these eigenvectors as its columns. The final structure is then determined by applying classical k-means to the rows of the matrix $\mathbf{U}$ [19]. The number of the eigenvectors, k, corresponds to the number of communities in the network.

3.3. Spectral Co-Clustering in Bipartite Networks

Bipartite spectral co-clustering is a technique for simultaneously clustering both rows and columns of a bipartite network [21,22]. The problem can be formulated as a trace maximization problem as follows:

$$\max_{\mathbf{Z} \in \mathbb{R}^{(n_1+n_2) \times k}} tr(\mathbf{Z}^{\top} \mathbf{B}_N \mathbf{Z}) = \max_{\mathbf{Z} \in \mathbb{R}^{(n_1+n_2) \times k}} tr\left(\begin{bmatrix} \mathbf{U}_L \\ \mathbf{U}_R \end{bmatrix}^{\top} \mathbf{B}_N \begin{bmatrix} \mathbf{U}_L \\ \mathbf{U}_R \end{bmatrix} \right), \tag{4}$$

where $\mathbf{B}_N = \begin{bmatrix} \mathbf{0} & \mathbf{A}_N^{12} \\ \mathbf{A}_N^{21} & \mathbf{0} \end{bmatrix}$, $\mathbf{Z} = \begin{bmatrix} \mathbf{U}_L \\ \mathbf{U}_R \end{bmatrix}$, $\mathbf{A}_N^{12} = \mathbf{D}_L^{-\frac{1}{2}} \mathbf{A}^{12} \mathbf{D}_R^{-\frac{1}{2}}$ with $D_{Lii} = \sum_j A_{ij}^{12}$, and $D_{Rjj} = \sum_i A_{ij}^{12}$. According to the Ky Fan theorem, the global optimum solution of Equation (4) is the matrix $\mathbf{Z}$ containing k eigenvectors that correspond to the largest eigenvalues of $\mathbf{B}_N$.

A more computationally efficient solution [21] to solve Equation (4) is to compute $\mathbf{U}_L$ and $\mathbf{U}_R$ as the matrices containing the left and right singular vectors that correspond to the largest k singular values of the matrix $\mathbf{A}_N^{12}$, respectively.

Another approach to solve Equation (4) is by first computing the symmetric rows and columns adjacency matrices, $\mathbf{A}_N^{12} \mathbf{A}_N^{12\top}$ and $\mathbf{A}_N^{21} \mathbf{A}_N^{21\top}$, and then solving two trace maximization problems, simultaneously, as follows [22]:

$$\begin{gathered} \max_{\mathbf{U}_L \in \mathbb{R}^{n_1 \times k}} tr\left(\mathbf{U}_L^{\top} \mathbf{A}_N^{12} \mathbf{A}_N^{12\top} \mathbf{U}_L\right) \;\; s.t \;\; \mathbf{U}_L^{\top}\mathbf{U}_L = \mathbf{I}_k \\ \text{and} \\ \max_{\mathbf{U}_R \in \mathbb{R}^{n_2 \times k}} tr\left(\mathbf{U}_R^{\top} \mathbf{A}_N^{21} \mathbf{A}_N^{21\top} \mathbf{U}_R\right) \;\; s.t \;\; \mathbf{U}_R^{\top}\mathbf{U}_R = \mathbf{I}_k, \end{gathered} \tag{5}$$

where $\mathbf{U}_L$ and $\mathbf{U}_R$ can be computed separately as the eigenvectors' matrices related to the largest k eigenvalues of $\mathbf{A}_N^{12} \mathbf{A}_N^{12\top}$ and $\mathbf{A}_N^{21} \mathbf{A}_N^{21\top}$, respectively.

3.4. Projection Distance between Subspaces

The projection distance between subspaces measures the distance between two subspaces of a vector space. In particular, it quantifies the distance between the orthogonal projections of a vector onto each of the two subspaces. Let $span(\mathbf{H}_1)$ and $span(\mathbf{H}_2)$ be two subspaces and their corresponding orthonormal basis sets are $\mathbf{H}_1 \in \mathbb{R}^{n\times k}$ and $\mathbf{H}_2 \in \mathbb{R}^{n\times k}$, respectively. The projection distance can be determined by the principal angles between the two subspaces. Let θ_i denote the ith principal angle between the two subspaces; then, the projection distance is defined as [23]:

$$\begin{aligned} d_p^2(span(\mathbf{H}_1),\, span(\mathbf{H}_2)) &= \sum_{i=1}^{k} \sin^2(\theta_i), \\ &= k - \sum_{i=1}^{k} \cos^2(\theta_i), \\ &= k - tr(\mathbf{H}_1\mathbf{H}_1^\top \mathbf{H}_2\mathbf{H}_2^\top). \end{aligned} \tag{6}$$

4. Community Detection in Multilayer Networks: A Unified Spectral Clustering Approach (ML-USCL)

4.1. Problem Formulation

Given a two-layer network, in order to determine a community, $\mathcal{C}_k$, two subsets of vertices are defined: (i) within-layer community subset, $\mathcal{C}_S = (V_m^{\mathcal{C}_k}, E_m^{\mathcal{C}_k})$, with $V_m^{\mathcal{C}_k} \subseteq V_m$ and $E_m^{\mathcal{C}_k} = E_m \cap (V_m^{\mathcal{C}_k} \times V_m^{\mathcal{C}_k})$, where $m \in \{1,2\}$, and (ii) across-layer community subset, $\mathcal{C}_B = (V_1^{\mathcal{C}_k}, V_2^{\mathcal{C}_k}, E_{12}^{\mathcal{C}_k})$ with $V_1^{\mathcal{C}_k} \subseteq V_1, V_2^{\mathcal{C}_k} \subseteq V_2$ and $E_{12}^{\mathcal{C}_k} = E_{12} \cap (V_1^{\mathcal{C}_k} \times V_2^{\mathcal{C}_k})$. A community can then be defined as $\mathcal{C}_k = \{\mathcal{C}_S, \mathcal{C}_B\}$, where it may include vertices from one or more layers in the network. In particular, $\mathcal{C}_k$ defines either a within-layer community when $\mathcal{C}_B = \varnothing$ or an across-layer community when $\mathcal{C}_B \neq \varnothing$.

In this paper, the objective is to partition a two-layer network into K disjoint communities. The goal is to find the low-rank embeddings of each layer to maximize the separability between communities while ensuring that these low-rank embeddings are consistent with the partitioning of the inter-layer graph. This objective is achieved by exploiting previous work in spectral clustering of single-layer networks and spectral co-clustering of bipartite networks. More precisely, intra-and inter-layer graphs are modeled as single-layer and bipartite graphs, respectively. The intra-layer graph encodes the interactions between nodes within the same layer, whereas the inter-layer graph encodes the interactions between nodes from both layers.

The proposed objective function can be expressed mathematically as:

$$\begin{aligned} \max_{\substack{\mathbf{U}_1\in\mathbb{R}^{n_1\times k_1}, \mathbf{U}_2\in\mathbb{R}^{n_2\times k_2}, \\ \mathbf{U}_L\in\mathbb{R}^{n_1\times k}, \mathbf{U}_R\in\mathbb{R}^{n_2\times k}}} \quad & \underbrace{tr(\mathbf{U}_1^\top \mathbf{A}_N^1 \mathbf{U}_1) + tr(\mathbf{U}_2^\top \mathbf{A}_N^2 \mathbf{U}_2)}_{\text{Within-Layer Normalized Cut}} \\ & + \underbrace{tr(\mathbf{U}_L^\top (\mathbf{A}_N^{12} {\mathbf{A}_N^{12}}^\top) \mathbf{U}_L) + tr(\mathbf{U}_R^\top (\mathbf{A}_N^{21} {\mathbf{A}_N^{21}}^\top) \mathbf{U}_R)}_{\text{Across-Layer Normalized Cut}} \\ & + \underbrace{\lambda_1 tr(\mathbf{U}_1 {\mathbf{U}_1}^\top \mathbf{U}_L {\mathbf{U}_L}^\top) + \lambda_2 tr(\mathbf{U}_2 {\mathbf{U}_2}^\top \mathbf{U}_R {\mathbf{U}_R}^\top)}_{\text{Regularization}} \\ \text{s.t} \quad & \mathbf{U}_1^\top \mathbf{U}_1 = \mathbf{I}_{k_1}, \mathbf{U}_2^\top \mathbf{U}_2 = \mathbf{I}_{k_2}, \mathbf{U}_L^\top \mathbf{U}_L = \mathbf{I}_k, \mathbf{U}_R^\top \mathbf{U}_R = \mathbf{I}_k. \end{aligned} \tag{7}$$

The proposed objective function is formulated such that the first two terms refer to the spectral clustering problem for Layers 1 and 2, respectively, where $\mathbf{A}_N^1 \in \mathbb{R}^{n_1\times n_1}$ and $\mathbf{A}_N^2 \in \mathbb{R}^{n_2\times n_2}$ are the symmetric normalized intra-layer adjacency matrices. The third and fourth terms refer to the bipartite spectral clustering problem. The last two terms define the spectral embedding similarity between the left and right subspaces of $\mathbf{A}_N^{12}$ and the low-rank subspaces of $\mathbf{A}_N^1$ and $\mathbf{A}_N^2$, respectively [24–26]. In particular, maximizing these two terms minimizes the projection subspace distance between $(\mathbf{U}_1, \mathbf{U}_L)$ and $(\mathbf{U}_2, \mathbf{U}_R)$, ensuring the consistency between the intra- and inter-layer partitions. k_1 and k_2 refer to the number of within-layer communities in Layers 1 and 2, respectively, whereas k refers to the number of across-layer communities. λ_1 and λ_2 are the regularization parameters.

4.2. Problem Solution

4.2.1. Initializing the Intra- and Inter-Layer Basis Matrices

Intra-layer basis matrices, $\mathbf{U}_1 \in \mathbb{R}^{n_1 \times k_1}$ and $\mathbf{U}_2 \in \mathbb{R}^{n_2 \times k_2}$, are initialized using Equation (3). The number of communities, k_1 and k_2, in each one of the layers is determined initially by the asymptotical surprise (AS) metric [27]. In particular, the AS metric is calculated for a range of possible community numbers, and the initial number of intra-communities is set to the number that achieves the maximum value of the AS metric. On the other hand, inter-layer basis matrices, $\mathbf{U}_L \in \mathbb{R}^{n_1 \times k}$ and $\mathbf{U}_R \in \mathbb{R}^{n_2 \times k}$, are initialized using Equation (5), where $k = min(k_1, k_2)$. The steps of initializing the basis matrices are oultined in Algorithm 1.

Algorithm 1 Initializing the intra- and inter-layer basis matrices

```
**Input:** $\mathbf{A}^1$, $\mathbf{A}^2$, $\mathbf{A}^{12}$, maximum number of communities ($C_m$).
**Output:** Initial ($\mathbf{U}_1$, $\mathbf{U}_2$, $\mathbf{U}_L$ and $\mathbf{U}_R$), $k_1$, $k_2$, $k$
Compute $\mathbf{A}_N^1$, $\mathbf{A}_N^2$ and $\mathbf{A}_N^{12}$
Compute eigendecomposition of $\mathbf{A}_N^1 = \mathbf{U}_1 \Lambda_1 \mathbf{U}_1^\top$ and $\mathbf{A}_N^2 = \mathbf{U}_2 \Lambda_2 \mathbf{U}_2^\top$
**for** $NOC = 2 : C_m$ **do**
    Apply k-means to $\mathbf{U}_1(:, 1 : NOC)$ to determine the first layer nodes clustering labels ($CL1_w$).
    Apply k-means to $\mathbf{U}_2(:, 1 : NOC)$ to determine the second layer nodes clustering labels ($CL2_w$).
    Compute asymptotical surprise (AS) for Layer 1: AS1(NOC)=AS($\mathbf{A}^1, CL1_w$)
    Compute asymptotical surprise (AS) for Layer 2: AS2(NOC)=AS($\mathbf{A}^2, CL2_w$)
**end for**
Find $k_1$ as $k_1 = \max_j AS1(j)$
Find $k_2$ as $k_2 = \max_j AS2(j)$
Set $k$ to $min(k_1, k_2)$
Compute $\mathbf{U}_L(:, 1 : k)$ and $\mathbf{U}_R(:, 1 : k)$ using Equation (5)
**return** $\mathbf{U}_1(:, 1 : k_1)$, $\mathbf{U}_2(:, 1 : k_2)$, $\mathbf{U}_L(:, 1 : k)$ and $\mathbf{U}_R(:, 1 : k)$, $k_1$, $k_2$ and $k$
```

4.2.2. Finding the Basis Matrices

As solving for the different variables in the proposed objective function jointly is not feasible, alternating maximization can be adopted to compute the variables, iteratively. Alternating maximization is a commonly used approach for solving optimization problems that involve multiple variables or constraints. In particular, the technique involves fixing one set of variables and computing the other set, alternating between the two until convergence [28,29].

The solution to the proposed problem in Equation (7) can be found using an alternating maximization scheme as follows:

- Update $\mathbf{U}_1$: By considering only the terms that contain $\mathbf{U}_1$, we obtain

$$\begin{aligned} &\max_{\mathbf{U}_1 \in \mathbb{R}^{n_1 \times k_1}} tr(\mathbf{U}_1^\top \mathbf{A}_N^1 \mathbf{U}_1) + \lambda_1 tr(\mathbf{U}_1 {\mathbf{U}_1}^\top \mathbf{U}_L {\mathbf{U}_L}^\top) \quad s.t \quad \mathbf{U}_1^\top \mathbf{U}_1 = \mathbf{I}_{k_1} \\ &= \max_{\mathbf{U}_1 \in \mathbb{R}^{n_1 \times k_1}} tr(\mathbf{U}_1^\top \mathbf{A}_{mod}^1 \mathbf{U}_1) \quad s.t \quad \mathbf{U}_1^\top \mathbf{U}_1 = \mathbf{I}_{k_1}, \end{aligned} \tag{8}$$

where $\mathbf{A}_{mod}^1 = \mathbf{A}_N^1 + \lambda_1 \mathbf{U}_L {\mathbf{U}_L}^\top$ is the modified normalized adjacency matrix representing Layer 1. The solution to this problem is similar to the classic spectral clustering formulation. In particular, the matrix $\mathbf{U}_1$ is computed through eigen decomposition

(ED) of $\mathbf{A}^1_{mod}$, where it consists of the k_1 eigenvectors that are associated with the k_1 largest eigenvalues of $\mathbf{A}^1_{mod}$.

- Update $\mathbf{U}_2$: By considering only the terms that contain $\mathbf{U}_2$ we obtain,

$$\begin{aligned}&\max_{\mathbf{U}_2 \in \mathbb{R}^{n_2 \times k_2}} tr(\mathbf{U}_2^\top \mathbf{A}^2_N \mathbf{U}_2) + \lambda_2 tr(\mathbf{U}_2 {\mathbf{U}_2}^\top \mathbf{U}_R {\mathbf{U}_R}^\top) \quad s.t \quad \mathbf{U}_2^\top \mathbf{U}_2 = \mathbf{I}_{k_2} \\ &= \max_{\mathbf{U}_2 \in \mathbb{R}^{n_2 \times k_2}} tr(\mathbf{U}_2^\top \mathbf{A}^2_{mod} \mathbf{U}_2) \quad s.t \quad \mathbf{U}_2^\top \mathbf{U}_2 = \mathbf{I}_{k_2},\end{aligned} \tag{9}$$

  where $\mathbf{A}^2_{mod} = \mathbf{A}^2_N + \lambda_2 \mathbf{U}_R {\mathbf{U}_R}^\top$ is the modified normalized adjacency matrix representing Layer 2. Similar to $\mathbf{U}_1$, $\mathbf{U}_2$ contains k_2 eigenvectors associated with the k_2 largest eigenvalues of $\mathbf{A}^2_{mod}$.

- Update $\mathbf{U}_L$: By considering only the terms that contain $\mathbf{U}_L$, we obtain

$$\begin{aligned}&\max_{\mathbf{U}_L \in \mathbb{R}^{n_1 \times k}} tr(\mathbf{U}_L^\top (\mathbf{A}^{12}_N {\mathbf{A}^{12}_N}^\top) \mathbf{U}_L) + \lambda_1 tr(\mathbf{U}_1 {\mathbf{U}_1}^\top \mathbf{U}_L {\mathbf{U}_L}^\top) \quad s.t \quad \mathbf{U}_L^\top \mathbf{U}_L = \mathbf{I}_k \\ &= \max_{\mathbf{U}_L \in \mathbb{R}^{n_1 \times k}} tr(\mathbf{U}_L^\top \bar{\mathbf{A}}_L \mathbf{U}_L) \quad s.t \quad \mathbf{U}_L^\top \mathbf{U}_L = \mathbf{I}_k,\end{aligned} \tag{10}$$

  where $\bar{\mathbf{A}}_L = \mathbf{A}^{12}_N {\mathbf{A}^{12}_N}^\top + \lambda_1 \mathbf{U}_1 {\mathbf{U}_1}^\top$ is referred to as the modified normalized adjacency matrix that represents the rows of the inter-layer graph and $\mathbf{U}_L$ can be computed also through eigendecomposition and it comprises the eigenvectors corresponding to the k largest eigenvalues of $\bar{\mathbf{A}}_L$.

- Update $\mathbf{U}_R$: By keeping all the terms that include only $\mathbf{U}_R$, we obtain

$$\begin{aligned}&\max_{\mathbf{U}_R \in \mathbb{R}^{n_2 \times k}} tr(\mathbf{U}_R^\top (\mathbf{A}^{21}_N {\mathbf{A}^{21}_N}^\top) \mathbf{U}_R) + \lambda_2 tr(\mathbf{U}_2 {\mathbf{U}_2}^\top \mathbf{U}_R {\mathbf{U}_R}^\top) \quad s.t \quad \mathbf{U}_R^\top \mathbf{U}_R = \mathbf{I}_k \\ &= \max_{\mathbf{U}_R \in \mathbb{R}^{n_2 \times k}} tr(\mathbf{U}_R^\top \bar{\mathbf{A}}_R \mathbf{U}_R) \quad s.t \quad \mathbf{U}_R^\top \mathbf{U}_R = \mathbf{I}_k,\end{aligned} \tag{11}$$

  where $\bar{\mathbf{A}}_R = {\mathbf{A}^{12}_N}^\top \mathbf{A}^{12}_N + \lambda_2 \mathbf{U}_2 {\mathbf{U}_2}^\top$ indicates the modified normalized adjacency matrix that represents the columns of the inter-layer graph, and $\mathbf{U}_R$ represents the eigenvector matrix that corresponds to the k largest eigenvalues of $\bar{\mathbf{A}}_R$.

As it can be seen from the update steps outlined above, each of the basis matrices is found jointly using both intra- and inter-layer adjacency information. In particular, $\mathbf{U}_1$ is the subspace corresponding to both its corresponding adjacency matrix and the Gram matrix defined by $\mathbf{U}_L$, i.e., $\mathbf{U}_L \mathbf{U}_L^\top$. Similar arguments can be made for the other basis matrices showing that they are learned to optimize the span for both within- and between-layer connectivity. After the basis matrices are computed, a set of intra- and inter-layer communities are determined by applying k-means to $\mathbf{U}_1$, $\mathbf{U}_2$, and $\mathbf{Z} = [\mathbf{U}_L; \mathbf{U}_R]$. The final community structure of the network is then determined by following the steps explained in Section 4.3.

4.3. Determining the Final Community Structure of the Network

After estimating a set of within- and across-layer communities, we evaluate their quality as follows:

1. The quality or strength of each within- and across-layer community is measured using the communitude metric [30,31] in terms of the supra-adjacency matrix as:

$$Communitude(\mathcal{C}_k) = \frac{\frac{E_{in}^{\mathcal{C}_k}}{E} - (\frac{E_{in}^{\mathcal{C}_k} + E_{ex}^{\mathcal{C}_k}}{2E})^2}{(\frac{E_{in}^{\mathcal{C}_k} + E_{ex}^{\mathcal{C}_k}}{2E})^2 (1 - (\frac{E_{in}^{\mathcal{C}_k} + E_{ex}^{\mathcal{C}_k}}{2E})^2)}, \tag{12}$$

   where $E_{in}^{\mathcal{C}_k}$ is the sum of internal edges in a community, i.e., edges between the nodes in the same community or “within-community” edges; $E_{ex}^{\mathcal{C}_k}$ is the sum of external edges in a community, i.e., edges that connect nodes belonging to different communities or "between-community" edges; and $2E$ refers to the sum of all edges in the supra-adjacency. In fact, this quality function can be seen as an adapted form of the Z-score function, where it is normalized by the standard deviation of the fraction of the number of edges within the subgraph. The upper bound of the communitude metric is 1.

2. To identify the most significant communities in a network, the communities are ranked in descending order depending on their communitude values. Every node in the proposed ML-USCL is allowed to join one community, within- or across-layer community. In particular, the communitude values of both the within- or across-layer communities are compared; then, the community that scores a higher value is selected.
3. The final community structure of the two-layer MLN is then considered as the set of K communities, $\mathcal{C}_{2M} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$, that scores the largest communitude values.

The steps of the developed algorithm are summarized in Algorithm 2.

Algorithm 2 Detecting Community Structure in **M**ulti**l**ayer networks: **U**nified **S**pectral **Cl**ustering (ML-USCL)

```
Input: $\mathbf{A}^1$, $\mathbf{A}^2$, $\mathbf{A}^{12}$, $\lambda_1$, $\lambda_2$
Output: Within- and across-layer communities.
Initialize $\mathbf{U}_1$, $\mathbf{U}_2$, $\mathbf{U}_L$ and $\mathbf{U}_R$ as explained in Algorithm 1.
while not converge do
    Calculate $\mathbf{U}_1$ using Equation (8)
    Calculate $\mathbf{U}_2$ using Equation (9)
    Calculate $\mathbf{U}_L$ using Equation (10)
    Calculate $\mathbf{U}_R$ using Equation (11)
end while
Apply k-means on $\mathbf{U}_1$ and $\mathbf{U}_2$ to determine the within-layer communities.
Apply k-means on $Z = [\mathbf{U}_L; \mathbf{U}_R]$ to determine the across-layer communities.
Compute the communitude of the detected communities using Equation (12).
return Best quality communities.
```

4.4. Computational Complexity of ML-USCL

The order of complexity of the proposed ML-USCL depends on the specific implementation of the algorithm and the size of the input network. The main steps involved in the proposed ML-USCL include initializing the basis matrices, updating the basis matrices by performing eigenvalue decomposition, and applying k-means to get the within- and across-layer communities.

Let n_m be the number of nodes in the mth layer; then, initializing each one of the basis matrices requires eigenvalue decomposition of the intra- and inter-layer adjacency matrices, which has a complexity of $\mathcal{O}(n_m^3)$. During the initialization step, the number of communities in each of the intra-layer graphs is determined by calculating the AS metric over a range of communities, $\{2, \ldots, C_m\}$. The AS calculation requires $C_m\mathcal{O}(|E^m|)$, where $|E^m|$ refers to the number of edges in the m^{th} layer. For each iteration, the update of $\mathbf{U}_m$ has a complexity of $\mathcal{O}(n_m^3)$. Applying k-means to $\mathbf{U}_m$ to get the clustering labels requires $\mathcal{O}(n_m l_m^{'} k_m)$, where $l_m^{'}$ refers to the number iterations taken by the k-means.

Overall, the complexity of the proposed ML-USCL is dominated by the eigenvalue decomposition step. In particular, the order of complexity when full eigenvalue decomposition is calculated can be determined as $l_m\mathcal{O}(n_m^3)$, where l_m refers to the total number of iterations. However, this computational complexity can be reduced to $l_m\mathcal{O}(k_m n_m^2)$ by computing the leading k_m eigenvectors. Consequently, the order of complexity of the proposed ML-USCL can be considered as $l_m\mathcal{O}(k_m n_m^2)$ since the leading k_m eigenvectors are computed in each iteration.

5. Results and Discussion

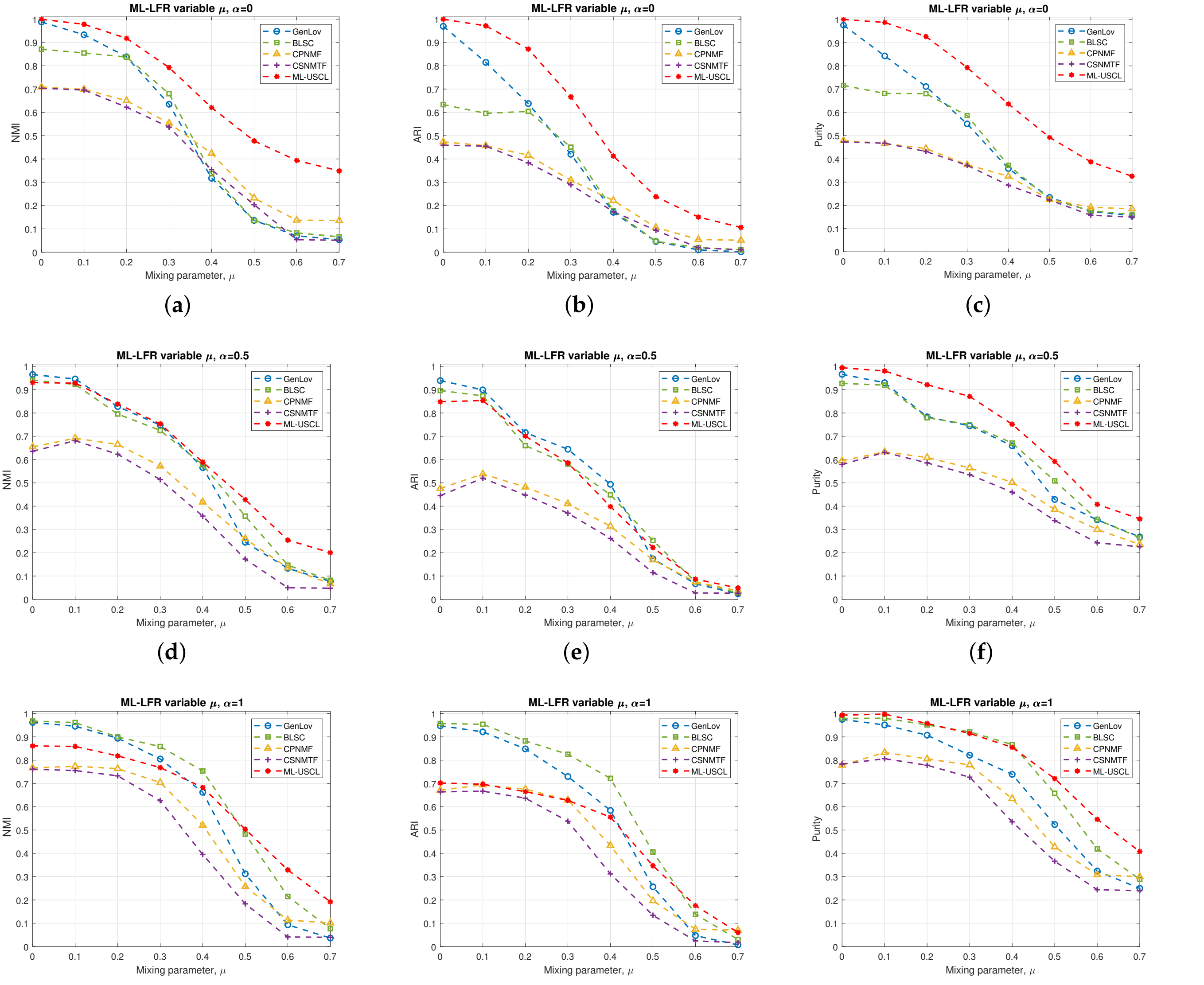
In this section, multiple experiments are conducted to evaluate the significance of the proposed approach. All experiments are performed on a standard Windows 10 Server with Intel (R) Core (TM) i7-9700 CPU @ 3.00GHz and 16GB RAM, MATLAB R2022b. The performance of the proposed ML-USCL is compared to other existing approaches, including block spectral clustering with inter-layer relations (BLSC), where $\beta = 1$ as suggested by the approach [16], generalized Louvain (GenLov) [32], collective NMF approaches [11], including CSNMTF and CPNMF. The input to the BLSC, GenLov, and collective NMF is the two-layer network directly, supra-adjacency, and the multiplex version of the network, respectively. The number of communities is determined using the asymptotical surprise for BLSC and the collective NMF methods, while it is self-optimized in GenLov. The value of the maximum number of communities, C_m, is determined based on the size of the network. For example, C_m can be set to 20 for small networks and to 100 for large networks. The quality of the network's final partition is evaluated using normalized mutual information (NMI) [33], adjusted Rand index (ARI) [34], and purity [35].

5.1. Simulated Networks

5.1.1. LFR Binary Simulated Networks

- LFR benchmark description: The Lancichinetti–Fortunato–Radicchi (LFR) benchmark [36,37] is a commonly used benchmark for evaluating the performance of community detection algorithms. In this experiment, the LFR benchmark is adopted to generate two-layer simulated networks, each with n nodes. The LFR benchmark uses a truncated power-law distribution to determine the community sizes. The parameters that control the community structure in the generated networks are (i) minimum degree, d_{min}, (ii) maximum degree, d_{max}, and (iii) mixing parameter, μ. The minimum and maximum values of the degree distribution (d_{min} and d_{max}, respectively) are chosen such that the average degree of the network is equal to $\langle d \rangle$. Within-layer graphs are generated with k_1 and k_2 communities, whereas the across-layer graph is generated such that the ratio of the across-layer communities to the total number of communities is greater than or equal to α, where $\alpha \in [0, 1]$. These selected communities are randomly combined with each other to create an across-layer community. The connection density within communities is set to $(1 - \mu)\langle d \rangle$ and between communities to $\mu\langle d \rangle$. The parameter μ controls the degree of inter-community connectivity, where a low μ value results in a strong community structure with few inter-community links, while a high μ value results in a weaker community structure with more inter-community links. On the other hand, α controls the percent of the across-layer communities. As α increases, the networks tend to have more across-layer communities.
- Experiment: In this experiment, two-layer unweighted networks with $n_1 = n_2 = 200$ are created. The parameters of the generated networks are $d_{min} = 8$, $d_{max} = 20$, $\mu = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$, and $\alpha = \{0.0, 0.5, 1.0\}$. To assess the efficacy of various algorithms in detecting community structure, a comparative analysis is carried out as the mixing parameter increases, i.e., the noise increases. Figure 1 shows a comparison between the different approaches in recovering the structure of two-layer LFR networks. Figure 1a–c reflect the performance of the different approaches when the two-layer network consists of within-layer communities only, i.e., $\alpha = 0$. As it can be noticed from the figures, the proposed ML-USCL outperforms the other methods significantly over the different values of the mixing parameter, μ, with respect to all of the evaluation metrics. In Figure 1d–f, where $\alpha = 0.5$, the two-layer networks comprises both intra- and inter-layer communities, the proposed ML-USCL exceeds the other algorithms in terms of the purity metric, and its performance is comparable to GenLov and BLSC with respect to the NMI and ARI metrics. Yet, the proposed ML-USCL performs better than both methods as the value of μ increases, which reflects its robustness. In Figure 1g–i, the networks consist of across-layer communities only. As illustrated in the figures, the proposed ML-USCL achieves higher scores of purity over

the range of μ, whereas GenLov and BLSC achieve better performance in terms of NMI and ARI. This improvement in the performance of GenLov and BLSC compared to the networks with $\alpha = 0$ and $\alpha = 0.5$ is due to the fact that the networks consist of larger communities. However, the performance of both methods is inferior to ML-USCL for $\mu > 0.4$, which indicates that BLSC and GenLov are not robust to noise.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 1. Comparison conducted among the various methods to evaluate their effectiveness in detecting the community structure. of LFR benchmark binary networks in terms of NMI, ARI, and purity with $\alpha = \{0, 0.5, 1\}$ and variable mixing parameter μ: (**a–c**) $\alpha = 0$; (**d–f**) $\alpha = 0.5$; (**g–i**) $\alpha = 1$.

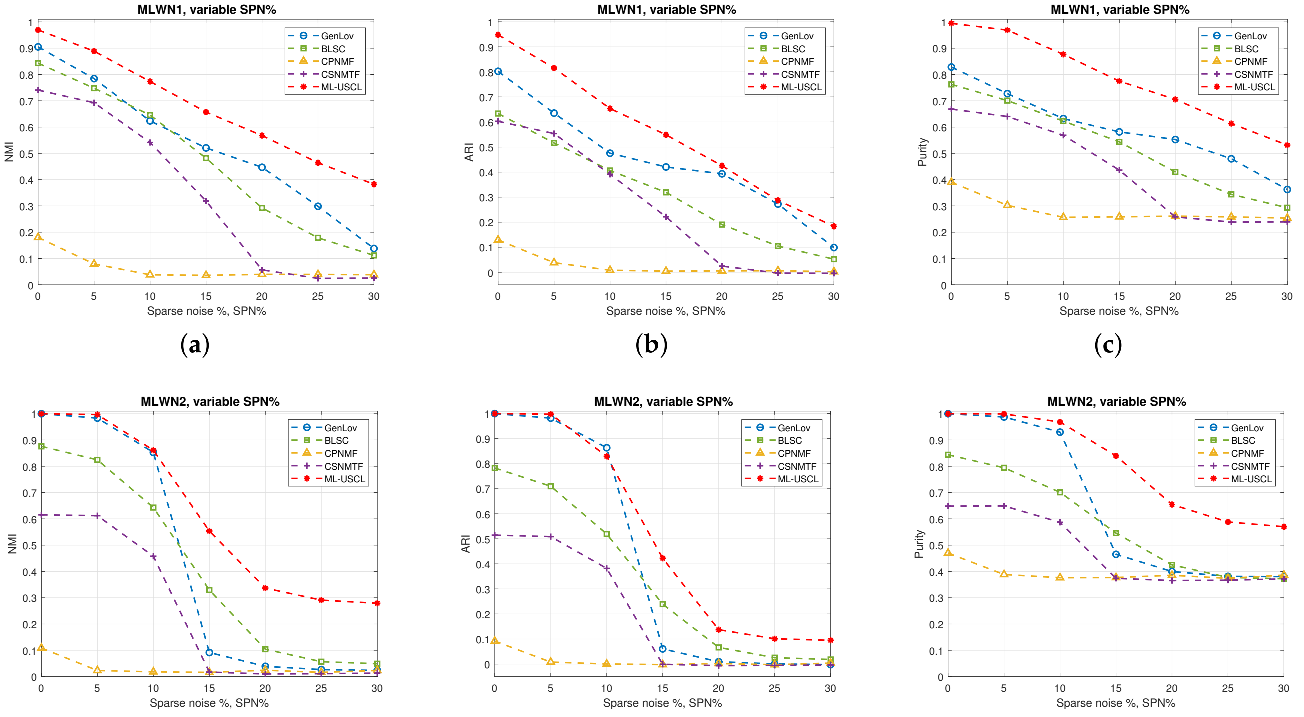
5.1.2. Weighted Simulated Networks

- Weighted network description: The two-layer simulated weighted MLNs are generated from a truncated Gaussian distribution in the range of $[0, 1]$. The networks are generated based on the parameters (μ_w, σ_w) and (μ_b, σ_b), which refer to the mean and standard deviation of edge weights within and between communities, respectively. Several two-layer MLNs are generated by varying the ground-truth structure, including the number of communities (NOCs). Furthermore, a percentage of sparse noise (SPN%) is randomly introduced into the networks to assess the algorithms' ability to handle noise.

- Experiment: In this experiment, multilayer weighted networks (MLWNs) with 2 layers and 100 nodes per layer are generated. The parameters of the constructed MLWNs are reported in Table 1. Three different MLWNs are generated with different ground-truth communities and varying sparse noise levels, SPN% $= \{0\%, 5\%, 10\%, 15\%, 20\%, 25\%, 30\%\}$. The proposed ML-USCL is compared to the other algorithms, and the results are shown in Figure 2. It is evident from Figure 2 the superiority of ML-USCL compared to the other algorithms in weighted networks in terms of all metrics. Moreover, the proposed approach exhibits robustness to the addition of sparse noise, with the ability to accurately detect the community structure even as the percentage of added sparse noise, SPN%, increases.

Table 1. Parameters of the simulated weighted two-layer MLNs.

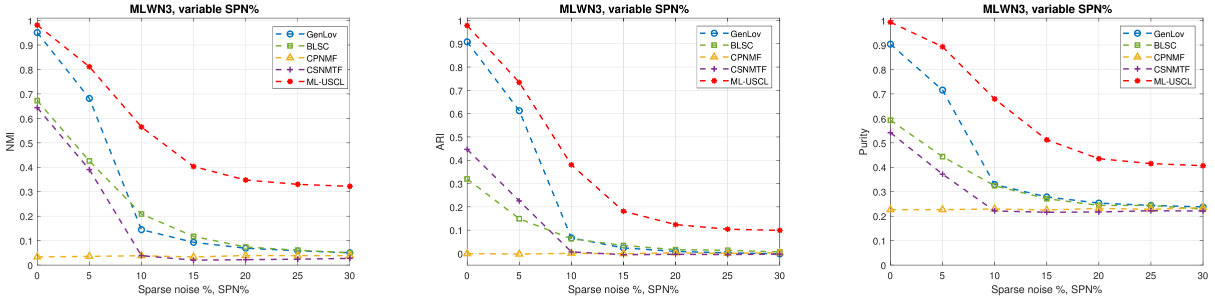
Network	n_1, n_2	NOC	Ground Truth: Cluster (Nodes in the Cluster)	$\mu_w, \sigma_w, \mu_b, \sigma_b$
MLWN 1	100, 100	7	**Graph1**: $\mathcal{C}_1^1$(1–20), $\mathcal{C}_2^1$(21–40), $\mathcal{C}_3^1$(41–60), $\mathcal{C}_4^1$(61–100)	$\mathcal{G}_1$: (0.5, 0.3, 0.3, 0.2)
			Graph2: $\mathcal{C}_1^2$(1–20), $\mathcal{C}_2^2$(21–35), $\mathcal{C}_3^2$(36–50), $\mathcal{C}_4^2$(51–75), $\mathcal{C}_5^2$(76–100)	$\mathcal{G}_2$: (0.5, 0.2, 0.2, 0.2)
			Ground truth: $\mathcal{C}_1^{12} = \{C_1^1, C_1^2\}$, $\mathcal{C}_2^{12} = \{C_2^1, C_2^2\}$, $\mathcal{C}_3^1$, $\mathcal{C}_4^1$,$\mathcal{C}_3^2$, $\mathcal{C}_4^2$, $\mathcal{C}_5^2$	$\mathcal{G}_{12}$: (0.8, 0.2, 0.3, 0.1)
MLWN 2	100, 100	4	**Graph1**: $\mathcal{C}_1^1$(1–40), $\mathcal{C}_2^1$(41–100)	$\mathcal{G}_1$: (0.5, 0.4, 0.3, 0.2)
			Graph2: $\mathcal{C}_1^2$(1–30), $\mathcal{C}_2^2$(31–60), $\mathcal{C}_3^2$(61–100)	$\mathcal{G}_2$: (0.5, 0.1, 0.2, 0.2)
			Ground truth: $\mathcal{C}_1^{12} = \{C_1^1, C_1^2\}$, $\mathcal{C}_2^1$, $\mathcal{C}_2^2$, $\mathcal{C}_3^2$	$\mathcal{G}_{12}$: (0.5, 0.2, 0.3, 0.1)
MLWN 3	100, 100	8	**Graph1**: $\mathcal{C}_1^1$(1–20), $\mathcal{C}_2^1$(21–40), $\mathcal{C}_3^1$(41–60), $\mathcal{C}_4^1$(61–100)	$\mathcal{G}_1$: (0.5, 0.4, 0.3, 0.2)
			Graph2: $\mathcal{C}_1^2$(1–15), $\mathcal{C}_2^2$(16–35), $\mathcal{C}_3^2$(36–50), $\mathcal{C}_4^2$(51–75), $\mathcal{C}_5^2$(76–100)	$\mathcal{G}_2$: (0.5, 0.1, 0.2, 0.2)
			Ground truth: $\mathcal{C}_1^{12} = \{C_1^1, C_1^2\}$, $\mathcal{C}_2^1$, $\mathcal{C}_3^1$, $\mathcal{C}_4^1$, $\mathcal{C}_2^2$, $\mathcal{C}_3^2$, $\mathcal{C}_4^2$, $\mathcal{C}_5^2$	$\mathcal{G}_{12}$: (0.5, 0.2, 0.3, 0.1)



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 2. *Cont.*



(g) **(h)** **(i)**

Figure 2. Comparison conducted among the various methods to evaluate their effectiveness in recovering the community structure of the multilayer weighted networks (MLWNs) from Table 1 in terms of NMI, ARI, and purity with different levels of added sparse noise percent (SPN%): (**a**–**c**) MLWN1; (**d**–**f**) MLWN2; (**g**–**i**) MLWN3.

5.2. Scalability Comparison

To evaluate the scalability of the proposed ML-USCL, we constructed a set of weighted multilayer networks with varying sizes. The network sizes ranged from 32 to 8192 on a logarithmic scale. For each network, the within- and between-community edges were randomly selected from a truncated Gaussian distribution within the range of $[0, 1]$ using the following parameters: $\mu_w^1 = 0.5$, $\sigma_w^1 = 0.1$, $\mu_b^1 = 0.3$, $\sigma_b^1 = 0.2$, $\mu_w^2 = 0.7$, $\sigma_w^2 = 0.1$, $\mu_b^2 = 0.2$, $\sigma_b^2 = 0.2$, $\mu_w^{12} = 0.5$, $\sigma_w^{12} = 0.1$, $\mu_b^{12} = 0.3$, and $\sigma_b^{12} = 0.2$, where the superscripts refer to the within- and across-layer graphs. Each network consisted of two equal-sized communities: $\mathcal{C}_1^1$ and $\mathcal{C}_2^1$ in Layer 1, and $\mathcal{C}_1^2$ and $\mathcal{C}_2^2$ in Layer 2. The community structure of the multilayer network comprised $\mathcal{C}_1^{12} = \mathcal{C}_1^1, \mathcal{C}_1^2$, $\mathcal{C}_2^1$, and $\mathcal{C}_2^2$. The number of communities was specified as an input for all the algorithms. The run time of the different algorithms was measured as the network size varied, and the results are displayed in Figure 3. Figure 3 illustrates that the run time of the different methods exhibits a log-linear relationship as the number of nodes increases. Moreover, the proposed ML-USCL performs better than CSNMTF as the number of nodes grows, and it is comparable to BLSC and CPNMF. However, GenLov exhibits a faster run time compared to ML-USCL. The order of complexity of the different algorithms is given in Table 2. Nonetheless, as shown by the different experiments, the proposed algorithm maintains good performance compared to all other methods in terms of detecting an accurate community structure.

Table 2. Computational complexity of the different methods: M is the number of layers, n is the total number of nodes in the MLN, n_m is the number of nodes in the mth layer, l_m^S is the maximum number of iterations required by CSNMTF, l_m^P is the maximum number of iterations required by CPNMF, l_m is the maximum number of iterations required by ML-USCL, K is the number of communities in the MLN, and k_m is the number of communities in the mth layer.

Method	Computational Complexity
GenLov	$\mathcal{O}(n \log n)$
BLSC	$\mathcal{O}(K(\sum_{m=1}^{M} n_m)^2)$
CSNMTF	$Ml_m^S\mathcal{O}(k_m n_m^2)$
CPNMF	$Ml_m^P\mathcal{O}(k_m n_m^2)$
ML-USCL	$Ml_m\mathcal{O}(k_m n_m^2)$

5.3. Regularization Parameters Selection

The proposed ML-USCL incorporates two regularization parameters, namely λ_1 and λ_2. Both parameters penalize the similarity between the orthonormal subspaces within and across layers. The impact of these regularization parameters on the algorithm's performance is investigated through experimental validation. We observed that the selection of

λ_1 and λ_2 depends on the characteristics of the multilayer network under examination. If the network primarily consists of within-layer communities, smaller values of the regularization parameters are advised. Conversely, if the network comprises predominantly across-layer communities or both within- and across-layer communities, larger values of the parameters are recommended. In the proposed ML-USCL, we search for the best λ_1 and λ_2 jointly over a grid of [0.1–10].

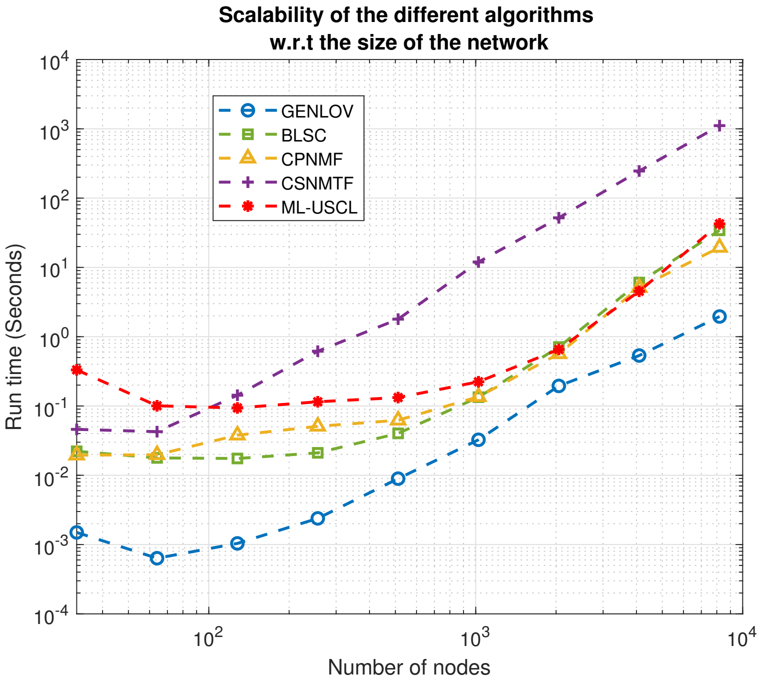


Figure 3. Scalability comparison between the different methods.

5.4. Real-World Networks

To evaluate the effectiveness of the proposed method and compare it with other algorithms in identifying the community structure in real-world networks, four real-world networks are considered, as shown in Table 3. A brief description of these networks is given in the following section.

Table 3. Description of the two-layer real networks.

Network	Nodes per Layer	Number of Edges	Average Degree	NOC
Lazega	71	4624	65.13	3
MIT	87	2790	32.06	6
C. Elegans	279	7266	27.03	12
Cora	292	19,044	10.80	3
Coil20	1440	8,294,400	3611.4	20
UCI	2000	1,011,534	478.99	12

5.4.1. Networks Description

1. Lazega law-firm network (https://manliodedomenico.com/data.php) (accessed on 20 June 2023) [38] is a multilayer network that represents the interactions between 71 partners and associates in a corporate law partnership, where each layer corresponds to a specific type of interaction among the individuals. The intra-layer graphs capture co-work and advice relationships, while the inter-layer graph reflects friendship relationships. The dataset also includes seven attributes that can be used to evaluate the quality of the detected communities. In this study, the ground-truth community structure is determined based on the office location attribute, which could be either Boston, Hartford, or Providence.

2. MIT Reality Mining (http://reality.media.mit.edu/download.php) (accessed on 20 June 2023) [39] network depicts various modes of mobile phone communication among 87 users, with edges indicating physical location, Bluetooth scans, and phone calls. The network (https://github.com/VGligorijevic/NF-CCE/tree/master/data/nets) (accessed on 20 June 2023) is constructed as a two-layer network, where the intra-layer graphs represent the physical location and Bluetooth scans, and the inter-layer graph represents phone call interactions. Further details on the construction of the network can be found in [40]. The ground-truth community structure in this network corresponds to the affiliations of the users.
3. The C. Elegans network (https://manliodedomenico.com/data.php) (accessed on 20 June 2023) [8,41] is a multilayer network that depicts the synaptic junctions, including electric and chemical monadic and polyadic, between neurons in the C. Elegans nervous system. The network comprises 279 neurons and each neuron is grouped into different categories such as bodywall, mechanosensory, and head motor neurons. These categories can be considered as the ground-truth structure. In the constructed two-layer network, intra-layer edges denote the monadic and polyadic synaptic junctions among neurons, whereas inter-layer edges represent the electric junctions.
4. Cora (https://people.cs.umass.edu/~mccallum/data.html) (accessed on 20 June 2023) data set is a subset of the Cora bibliographic data set. The Cora MLN consists of 292 nodes that refer to research papers. The intra-layer edges reflect the title and abstract similarities between the different research papers and the inter-layer edges model the citation relationships between them. The clusters in the network correspond to the research fields, namely data mining, natural language processing, and robotics.
5. COIL20 (https://www.cs.columbia.edu/CAVE/software/softlib/) (accessed on 20 June 2023) is a data set comprising 1440 images obtained from the Columbia object image library where intra- and inter-layers represent different image features. Intra-layer graphs represent the local binary patterns (LBPs) and Gabor features, whereas inter-layer graphs represent the intensity feature. The data set consists of 20 communities, each referring to a group of related images.
6. UCI (https://archive.ics.uci.edu/ml/datasets/Multiple+Features) (accessed on 20 June 2023) [42] consists of features extracted from handwritten digits (0–9) obtained from a collection of Dutch utility maps. The dataset contains a total of 2000 digit patterns, with 200 patterns per digit. These patterns are represented using different sets of features, including Fourier coefficients of the character shapes, profile correlations, and Karhunen–Lo*é*ve coefficients.

5.4.2. Experiments

The accuracy of the proposed ML-USCL in uncovering the community structure of the four real-world networks is evaluated and compared with other existing methods. The results are reported in Table 4. Based on the table, it is evident that the proposed ML-USCL for community detection in two-layer MLNs shows significant improvement over other algorithms. The evaluation of the community structure is performed individually for each layer and the proposed approach exhibits better performance than the other algorithms, as it achieves superior results in at least one quality metric in both layers. This indicates that the proposed approach is highly effective in identifying communities in MLNs. Moreover, the run time required for each one of the methods is presented in Table 5, where the proposed ML-USCL is faster than CSNMTF and CPNMF, comparable to BLSC and slower than GenLov. Nevertheless, ML-USCL exceeds the other algorithms in performance. These results suggest that the proposed approach can be a valuable tool in various applications that require the identification of communities in MLNs, such as social, biological, phone, and citation network analysis. Overall, the proposed approach has the potential to advance the field of community detection in MLNs and enable more accurate and efficient analysis of complex systems.

Table 4. Community detection performance comparison for real-world multilayer networks.

Network	Metric	BLSC	GenLov	CSNMTF	CPNMF	ML-USCL
Lazega	**NMI**	0.5129	0.5816	0.5732	0.6332	0.8943
Layer1	**ARI**	0.3524	0.4724	0.4445	0.4731	0.9515
	Purity	0.9155	0.9296	0.9296	0.9437	0.9577
Lazega	**NMI**	0.4858	0.5816	0.5732	0.6332	0.7507
Layer2	**ARI**	0.3596	0.4724	0.4445	0.4731	0.8482
	Purity	0.9014	0.9296	0.9296	0.9437	0.9437
MIT	**NMI**	0.1319	0.1808	0.3197	0.3358	0.4647
Layer1	**ARI**	0.0326	0.0030	0.2193	0.2803	0.3498
	Purity	0.4828	0.4943	0.6092	0.6322	0.6782
MIT	**NMI**	0.4199	0.5251	0.3197	0.3358	0.4760
Layer2	**ARI**	0.3092	0.3889	0.2193	0.2803	0.3855
	Purity	0.6437	0.6897	0.6092	0.6322	0.7011
C. Elegans	**NMI**	0.4614	0.4211	0.4352	0.4710	0.4735
Layer1	**ARI**	0.2429	0.2652	0.1771	0.2138	0.3714
	Purity	0.5167	0.4796	0.5613	0.5502	0.5279
C. Elegans	**NMI**	0.4404	0.4198	0.4352	0.4710	0.4669
Layer2	**ARI**	0.1988	0.2703	0.1771	0.2138	0.3620
	Purity	0.4870	0.4833	0.5613	0.5502	0.5130
Cora	**NMI**	0.1919	0.5634	0.5136	0.2315	0.5700
Layer1	**ARI**	0.0276	0.5615	0.5306	0.1037	0.5059
	Purity	0.4692	0.8938	0.8699	0.5205	0.8390
Cora	**NMI**	0.124	0.6420	0.5136	0.2315	0.8351
Layer2	**ARI**	0.0163	0.6502	0.5306	0.1037	0.8815
	Purity	0.4075	0.9247	0.8699	0.5205	0.9692
Coil20	**NMI**	0.1921	0.3962	0.7458	0.0429	0.7719
Layer1	**ARI**	0.0132	0.1307	0.5146	0	0.6174
	Purity	0.1438	0.1500	0.6743	0.0993	0.7653
Coil20	**NMI**	0	0.3917	0.7458	0.0429	0.7646
Layer2	**ARI**	0	0.1301	0.5146	0	0.6105
	Purity	0.0500	0.1500	0.6743	0.0993	0.7188
UCI	**NMI**	0.7613	0.7757	0.8006	0.7566	0.8064
Layer1	**ARI**	0.6761	0.6383	0.7427	0.6023	0.7722
	Purity	0.8170	0.6615	0.8775	0.7600	0.8990
UCI	**NMI**	0.7756	0.7878	0.8006	0.7566	0.8050
Layer2	**ARI**	0.6863	0.6524	0.7427	0.6023	0.7441
	Purity	0.8210	0.6680	0.8775	0.7600	0.8880

Table 5. Run time taken by the different methods for the real-world MLNs.

Network	GenLov	BLSC	CSNMTF	CPNMF	ML-USCL
Lazega	0.0250	0.2549	2.4958	0.5251	0.2155
MIT	0.0108	0.1873	4.9985	0.7384	0.2594
C. Elegans	0.0184	1.7216	81.6031	11.3486	0.9102
Cora	0.0310	0.6031	14.2655	0.7248	0.5368
Coil20	0.9188	2.8096	155.743	3.0271	5.3361
UCI	0.8801	6.4001	394.735	11.255	29.5577

6. Conclusions

Community detection in multilayer networks is an active research area with various challenges and opportunities. The structure of multilayer networks provides additional information that can be used to enhance the accuracy and interpretability of community detection methods. In this article, a unified spectral-clustering-based community detection method for two-layer MLNs is introduced. The task of identifying the community structure in two-layer MLNs is expressed as an optimization problem, in which the normalized cut is minimized for each layer while also considering the normalized cut for the bipartite network. This optimization is performed concurrently with regularization terms that ensure the coherence of the community structures both within and across layers.

Multiple experiments have been conducted to evaluate the effectiveness of the proposed approach for community detection in two-layer unweighted and weighted simulated and real-world MLNs. These experiments demonstrate the efficiency and accuracy of the proposed ML-USCL in detecting the community structure in two-layer MLNs. In addition, ML-USCL is robust to noise compared to existing approaches. Finally, the ability to use the same objective function for both weighted and unweighted networks, while being robust to noise and outliers, makes the proposed method applicable to a wide range of MLNs.

For future work, we will focus on generalizing the proposed ML-USCL and addressing its limitations. In particular, the proposed approach will be extended to handle MLNs with more than two layers. The extension will be developed by first constructing a multidimensional array or tensor that represents the multilayer network and then applying tensor decomposition to reveal the underlying communities in the network. Finally, we would like to point out that the proposed approach in its current formulation can be adopted to detect the community structure in heterogeneous MLNs, i.e., nodes in the different layers refer to different objects. Future work will perform more experiments to validate the extension of the proposed ML-USCL in heterogeneous MLNs.

Author Contributions: Conceptualization, E.A.-s. and S.A.; methodology, E.A.-s. and S.A.; software, E.A.-s. and S.A.; validation, E.A.-s. and S.A.; formal analysis, E.A.-s. and S.A.; validation, E.A.-s. and S.A.; investigation, E.A.-s. and S.A.; resources, E.A.-s. and S.A.; data curation, E.A.-s. and S.A.; writing—original draft preparation, E.A.-s. and S.A.; writing—review and editing, E.A.-s. and S.A.; visualization, E.A.-s. and S.A.; supervision, E.A.-s. and S.A.; project administration, E.A.-s. and S.A.; funding acquisition, E.A.-s. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Jordan University of Science and Technology under Research Grant Number 20220277 and the National Science Foundation under CCF-2006800.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank our colleague Abdullah Karaaslanli for providing us with the multilayer LFR network code.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Boccaletti, S.; Latora, V.; Moreno, Y.; Chavez, M.; Hwang, D.U. Complex networks: Structure and dynamics. *Phys. Rep.* **2006**, *424*, 175–308. [CrossRef]
2. Al-Sharoa, E.; Al-Khassaweneh, M.; Aviyente, S. Tensor based temporal and multilayer community detection for studying brain dynamics during resting state fMRI. *IEEE Trans. Biomed. Eng.* **2018**, *66*, 695–709. [CrossRef] [PubMed]
3. Kivelä, M.; Arenas, A.; Barthelemy, M.; Gleeson, J.P.; Moreno, Y.; Porter, M.A. Multilayer networks. *J. Complex Netw.* **2014**, *2*, 203–271. [CrossRef]
4. Papakostas, D.; Basaras, P.; Katsaros, D.; Tassiulas, L. Backbone formation in military multi-layer ad hoc networks using complex network concepts. In Proceedings of the MILCOM 2016—2016 IEEE Military Communications Conference, Baltimore, MD, USA, 1–3 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 842–848.
5. Berlingerio, M.; Coscia, M.; Giannotti, F. Finding and characterizing communities in multidimensional networks. In Proceedings of the 2011 international conference on advances in social networks analysis and mining, Kaohsiung, Taiwan, 25–27 July 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 490–494.
6. Burgess, M.; Adar, E.; Cafarella, M. Link-prediction enhanced consensus clustering for complex networks. *PLoS ONE* **2016**, *11*, e0153384. [CrossRef] [PubMed]

7. 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] [PubMed]
8. De Domenico, M.; Porter, M.A.; Arenas, A. MuxViz: A tool for multilayer analysis and visualization of networks. *J. Complex Netw.* **2015**, *3*, 159–176. [CrossRef]
9. Kuncheva, Z.; Montana, G. Community detection in multiplex networks using locally adaptive random walks. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, Paris, France, 25–28 August 2015; pp. 1308–1315.
10. Alimadadi, F.; Khadangi, E.; Bagheri, A. Community detection in facebook activity networks and presenting a new multilayer label propagation algorithm for community detection. *Int. J. Mod. Phys. B* **2019**, *33*, 1950089. [CrossRef]
11. Gligorijević, V.; Panagakis, Y.; Zafeiriou, S. Non-negative matrix factorizations for multiplex network analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 928–940. [CrossRef] [PubMed]
12. Ma, X.; Dong, D.; Wang, Q. Community detection in multi-layer networks using joint non-negative matrix factorization. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 273–286. [CrossRef]
13. Al-Sharoa, E.M.; Aviyente, S. Community Detection in Fully-Connected Multi-layer Networks Through Joint Nonnegative Matrix Factorization. *IEEE Access* **2022**, *10*, 43022–43043. [CrossRef]
14. Zhang, H.; Wang, C.D.; Lai, J.H.; Yu, P.S. Modularity in complex multilayer networks with multiple aspects: A static perspective. *Appl. Inform.* **2017**, *4*, 7. [CrossRef]
15. Pramanik, S.; Tackx, R.; Navelkar, A.; Guillaume, J.L.; Mitra, B. Discovering community structure in multilayer networks. In Proceedings of the 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, 19–21 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 611–620.
16. Chen, C.; Ng, M.; Zhang, S. Block spectral clustering for multiple graphs with inter-relation. *Netw. Model. Anal. Health Inform. Bioinform.* **2017**, *6*, 8. [CrossRef]
17. De Domenico, M.; Lancichinetti, A.; Arenas, A.; Rosvall, M. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Phys. Rev. X* **2015**, *5*, 011027. [CrossRef]
18. Yang, Z.; Algesheimer, R.; Tessone, C.J. A comparative analysis of community detection algorithms on artificial networks. *Sci. Rep.* **2016**, *6*, 30750. [CrossRef]
19. Von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [CrossRef]
20. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On spectral clustering: Analysis and an algorithm. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 9–14 December 2002; pp. 849–856.
21. Dhillon, I.S. Co-clustering documents and words using bipartite spectral graph partitioning. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001; ACM: New York, NY, USA, 2001; pp. 269–274.
22. Mirzal, A.; Furukawa, M. Eigenvectors for clustering: Unipartite, bipartite, and directed graph cases. In Proceedings of the 2010 International Conference on Electronics and Information Engineering, Kyoto, Japan, 1–3 August 2010; IEEE: Piscataway, NJ, USA, 2010; Volume 1, pp. V1-303–V1-309.
23. Hamm, J.; Lee, D.D. Grassmann discriminant analysis: A unifying view on subspace-based learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 376–383.
24. Dong, X.; Frossard, P.; Vandergheynst, P.; Nefedov, N. Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds. *IEEE Trans. Signal Process.* **2013**, *62*, 905–918. [CrossRef]
25. Kumar, A.; Rai, P.; Daume, H. Co-regularized multi-view spectral clustering. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 1413–1421.
26. Chi, Y.; Song, X.; Zhou, D.; Hino, K.; Tseng, B.L. Evolutionary spectral clustering by incorporating temporal smoothness. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, 12–15 August 2007; pp. 153–162.
27. Traag, V.; Aldecoa, R.; Delvenne, J. Detecting communities using asymptotical surprise. *Phys. Rev. E* **2015**, *92*, 022816. [CrossRef]
28. Bezdek, J.C.; Hathaway, R.J. Some notes on alternating optimization. In Proceedings of the Advances in Soft Computing—AFSS 2002: 2002 AFSS International Conference on Fuzzy Systems, Calcutta, India, 3–6 February 2002; Springer: Cham, Switzerland, 2002; pp. 288–300.
29. Bezdek, J.C.; Hathaway, R.J. Convergence of alternating optimization. *Neural Parallel Sci. Comput.* **2003**, *11*, 351–368.
30. Miyauchi, A.; Kawase, Y. What is a network community? A novel quality function and detection algorithms. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 1471–1480.
31. Chakraborty, T.; Dalmia, A.; Mukherjee, A.; Ganguly, N. Metrics for community analysis: A survey. *ACM Comput. Surv.* **2017**, *50*, 54. [CrossRef]
32. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
33. Danon, L.; Diaz-Guilera, A.; Duch, J.; Arenas, A. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, *2005*, P09008. [CrossRef]
34. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [CrossRef]

35. Schütze, H.; Manning, C.D.; Raghavan, P. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; Volume 39.
36. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **2008**, *78*, 046110. [CrossRef]
37. Bródka, P. A method for group extraction and analysis in multilayer social networks. *arXiv* **2016**, arXiv:1612.02377.
38. Lazega, E. *The Collegial Phenomenon: The Social Mechanisms of Cooperation among Peers in a Corporate Law Partnership*; Oxford University Press on Demand: Oxford, UK, 2001.
39. Eagle, N.; Pentland, A.S.; Lazer, D. Inferring friendship network structure by using mobile phone data. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 15274–15278. [CrossRef] [PubMed]
40. Dong, X.; Frossard, P.; Vandergheynst, P.; Nefedov, N. Clustering with multi-layer graphs: A spectral perspective. *IEEE Trans. Signal Process.* **2012**, *60*, 5820–5831. [CrossRef]
41. Chen, B.L.; Hall, D.H.; Chklovskii, D.B. Wiring optimization can relate neuronal structure and function. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 4723–4728. [CrossRef] [PubMed]
42. Dua, D.; Graff, C. UCI Machine Learning Repository. 2017. Available online: http://archive.ics.uci.edu/ml (accessed on 20 June 2023).

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.