RESEARCH ARTICLE

Community Detection in Multiplex Networks Based on Orthogonal Nonnegative Matrix Tri-Factorization

MEIBY ORTIZ-BOUZA, (Graduate Student Member, IEEE),
AND SELIN AVIYENTE, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA
Corresponding author: Selin Aviyente (aviyente@egr.msu.edu)

This work was supported in part by the National Science Foundation under Grant CCF-2006800 and in part by the Air Force Office of Scientific Research Grant FA9550-23-1-0224.

ABSTRACT

Networks are commonly used to model complex systems. The different entities in the system are represented by nodes of the network and their interactions by edges. In most real life systems, the different entities may interact in different ways necessitating the use of multiplex networks where multiple links are used to model the interactions. One of the major tools for inferring network topology is community detection. Although there are numerous works on community detection in single-layer networks, existing community detection methods for multiplex networks mostly learn a common community structure across layers and do not take the heterogeneity across layers into account. In this paper, we introduce a new multiplex community detection method that identifies communities that are common across layers as well as those that are unique to each layer. The proposed method, Multiplex Orthogonal Nonnegative Matrix Tri-Factorization, represents the adjacency matrix of each layer as the sum of two low-rank matrix factorizations corresponding to the common and private communities, respectively. Unlike most of the existing methods which require the number of communities to be pre-determined, the proposed method also introduces a two stage method to determine the number of common and private communities. The proposed algorithm is evaluated on synthetic and real multiplex networks, as well as for multiview clustering applications, and compared to state-of-the-art techniques.

INDEX TERMS

Multiplex networks, community detection, nonnegative matrix tri-factorization, eigengap, low-rank structure.

I. INTRODUCTION

Complex networks are usually used to represent many real world systems, ranging from social to biological ones [1], where the different agents and their relations are represented as the nodes and edges of the network, respectively. Traditional network models employ simple graphs, where there is a single edge between any two nodes. Thus, these models cannot capture multiple modes of interaction that may exist between the nodes. Recently, multiplex networks that represent multiple modes of interaction have been proposed. A multiplex network is a multilayer network where all layers share the same set of nodes with different topologies [2].

Multiplex networks have been used to model a variety of complex systems, including living organisms, human societies, transportation systems and critical infrastructures [3], [4].

Community detection is an important tool in network analysis, where communities are defined as groups of nodes that are more densely connected to each other than they are to the rest of the network. Detecting the community structure is useful for understanding the structure and function of complex networks. Community detection in networks is an ill-defined problem as there is no universal definition of what constitutes a “good” community. Numerous methods have been proposed for detecting the community structure of networks in single layer networks [5]. Among these, the most commonly used ones are spectral clustering [6], [7], methods based on statistical inference [8], approaches based...
of individuals will likely not work at the same company. Thus, in a situation like this, a given community will only be present in a subset of the layers, and different communities may be present in different subsets of layers.

In this paper, common communities are defined as communities that are observed in more than one layer, i.e., communities that are common across any subset of two or more layers, and private communities as communities that are unique to each layer. The problem of detecting common and private communities is then formulated using a novel framework titled Multiplex Orthogonal Nonnegative Matrix Tri-factorization (MX-ONMTF). In the proposed framework, each layer’s adjacency matrix is represented as the sum of two low-rank matrix factorizations corresponding to the common and private communities, respectively. The resulting optimization problem is solved using an iterative multiplicative update algorithm. The proposed approach also addresses the problem of determining the number of communities. Unlike most existing work, where the number of communities is determined through a greedy search, in this paper, a two-step approach is proposed. The proposed algorithm is first evaluated on synthetic benchmark multiplex networks with different numbers of layers, nodes, communities, noise levels, and inter-layer dependency probability. Next, the proposed method is applied to real networks including social and biological networks. Finally, the algorithm is evaluated for multiview clustering task, where the communities across all layers are assumed to be the same.

The rest of the paper is organized as follows. Section II presents a summary of related works. Section III provides background on multiplex networks and nonnegative matrix factorization based community detection methods. Sections IV and V present the proposed multiplex community detection algorithm and its convergence analysis, while Section VI illustrates results on both simulated and real networks. Finally, Section VII provides conclusions and discussion on future work.

II. RELATED WORKS

The method proposed in this paper belongs to the third class of algorithms, which operate directly on the multiplex network model. There are different types of algorithms that fall in this class: random walk, statistical generative network models, label propagation, objective function optimization, and Nonnegative Matrix Factorization (NMF).

Methods based on random walkers model the dynamic process on networks as random walks where the process is more likely to persist on the vertices in the same community and far less on the vertices in different communities. For instance, LART [19] is initialized by assigning each node in each layer to its own community. Hierarchical clustering is then used to merge nodes based on a distance matrix, and the partition with the highest multiplex modularity is chosen. In [20], Infomap, which is based on a compression of network flows, is proposed to identify communities within and across layers. However, Infomap tends to assign each physical node

| TABLE 1. Notation definitions. |
|---------------------------------|
| **Notations** | **Explanations** |
| G_i | i-th layer graph |
| n | Number of nodes |
| L | Number of layers |
| V | Set of nodes |
| E_l | Set of edges in layer l |
| A_l | Adjacency matrix for layer l |
| A_l(i, j) | i-th row and j-th column element of matrix A_l |
| L | Normalized Laplacian |
| ||F | Frobenius norm of a matrix A |
| k_l | Number of communities in layer l |
| k_c | Number of total common communities |
| k_p | Number of private communities in layer l |
| H ∈ R^n×k_c | Common communities membership matrix |
| H_l ∈ R^n×k_p | Private communities membership matrices |
| S_l ∈ R^k_c×k_p | Diagonal tri-factor matrices corresponding to the common communities |
| G_l ∈ R^{k_c×k_p} | Diagonal tri-factor matrices corresponding to the private communities |
| v_l | Community labels |
| L^{null}_l | Normalized Laplacian of the Erdős–Rényi null model |
| Δ_l | Diagonal matrix whose entries are the eigenvalues of the null model |
| δ | Threshold for the noise eigenvalues of the null model |
| U_l | Community membership matrix for each layer |
| X | Concatenation of U_l matrices used to cluster the communities per layer in Alg. 2 |
| F | Output of the hierarchical clustering algorithm |
| ⊕ | Hadamard product |
across layers to the same community, not differentiating the topological differences across layers.

Statistical methods use variants of Stochastic Block Model (SBM) to model the latent variables. Among these, multilayer stochastic block model (MLSBM) [25], [26], [27], [28], [29] model each layer’s adjacency matrix through a common community membership matrix, \( Z \), and a layer specific connectivity probability matrix \( B^m \), where the goal is to infer \( Z \). Consistency properties of various methods such as orthogonal linked matrix factorization (OLMF) [18], [25], [30] and spectral clustering under MLSBM have been investigated [25], [26], [27]. More recently, mixture multilayer stochastic block model (MMLSBM) has been proposed to model the heterogeneity of multiplex networks [31], [32], [33], [34]. MMLSBM assumes that there is a mixture of \( m \) latent network models and each network is sampled independently from this mixture of models with each of the \( m \) classes following SBM. While these methods provide some flexibility in modeling heterogeneous networks, they still make assumptions such that the layers can be clustered into subgroups where each subgroup has exactly the same community structure and connectivity. Similarly, Weighted Stochastic Block Model (WSBM) [35] has been proposed to detect common and private communities in heterogeneous weighted networks. Although this method addresses the heterogeneity of networks across layers, the method is limited to detecting only common communities that are shared by all layers, ignoring communities that may be shared by only a subset or different subset of layers. In [36], authors propose a generative model and an expectation maximization algorithm for community detection and link prediction in multilayer networks. Although the method allows for different connectivity patterns in each layer, the interdependence between layers is only taken into account for link prediction, while the layers are assumed to share a common community structure.

The third class of methods, Label Propagation Algorithms (LPA), is based on the intuition that a label can become dominant in a densely connected group of nodes but will have trouble crossing a sparsely connected region. In [37], an LPA-based method for community detection in multidimensional networks is proposed to identify communities and the subset of layers in which each of these communities is observed, simultaneously. However, this algorithm fails to detect communities that are private to each layer and communities that may be common among a small number of layers.

The fourth type of multiplex community detection methods is based on defining an objective function and identifying the community structure that maximizes/minimizes the objective function. For example, Generalized Louvain (GenLouvain) [21] uses an extended definition of modularity and is one of the fastest methods for community detection in multiplex networks. As GenLouvain assigns each node-layer tuple to its own community, it cannot identify common communities across layers. More recently, multiobjective genetic and evolutionary algorithms such as MultiMOGA [22] and MOEA/D-TS [38] have been used to jointly maximize the modularity of each layer and the similarity between the community structures across layers. These methods find a shared community structure across all layers, not differentiating communities that may be unique to each layer. In [39], extension of normalized cut to multiplex networks is proposed by constructing a block Laplacian matrix with each block corresponding to a layer. This method relies on selecting a parameter \( \beta \) that controls the consistency of the community structure across different layers.

The last class of methods is based on NMF which, because of its interpretability and good performance, has been broadly used for community detection in single-layer, multiplex, multilayer, and dynamic networks [40], [41], [42], [43]. In [44], Semi-Supervised joint Nonnegative Matrix Factorization (S2-jNMF) is proposed for detecting the common communities across layers in a multiplex network. A greedy search of dense subgraphs is performed and these subgraphs are used as \textit{a priori} information to create new adjacency matrices for each layer. In [45], a two-step approach is proposed, where first a nonnegative low dimensional feature representation of each layer is found using one of the four different NMF models. These community structures are then used to obtain a consensus community structure. Authors in [46] use NMF for detecting communities in multiplex social networks, where both unifying and coupling approaches are proposed. The unifying approach finds a common community structure by aggregating all layers, while the coupling approach finds mostly consistent community structures.

Most of the aforementioned NMF based methods find a common structure across all layers or for a majority of layers and do not consider cases where common communities may be present in different subsets of layers. Moreover, they do not detect private communities. These methods also require that the number of communities is provided \textit{a priori}.

### III. BACKGROUND

#### A. MULTIPLEX NETWORKS

Multiplex networks can be represented using a finite sequence of graphs \( \{G_l\} \), where \( l \in \{1, 2, \ldots, L\} \), \( G_l = (V, E_l, A_l) \) [47]. \( V \) is the set of nodes which is the same for all layers \( l \) and \( A_l \in \mathbb{R}^{n \times n} \) is the adjacency matrix for layer \( l \). In this paper, we use undirected (symmetric) weighted and binary adjacency matrices. For a weighted adjacency matrix, \( A_l(i, j) \in [0, 1] \), and for a binary adjacency matrix, \( A_l(i, j) \in \{0, 1\} \). A graph can also be represented by its normalized Laplacian matrix \( L = D^{-1/2} (D - A) D^{-1/2} \), with \( D \) being the diagonal degree matrix where the diagonal entries are \( D_{ii} = \sum_j A_{ij} \).

#### B. NONNEGATIVE MATRIX FACTORIZATION BASED COMMUNITY DETECTION

Methods based on Nonnegative matrix factorization (NMF) and its variants have been popular for community detection [48]. Compared to other community detection methods, these methods have some unique advantages including...
An ideal common community \( C \) in a multiplex network is defined as a subgraph with the same set of nodes for a subset of layers \( m \subset \{1, 2, \ldots, L\} \), where \(|m| > 1\).

**Definition 1.** An ideal common community \( C \) in a multiplex network \( \{G_l\} \) is defined as a subgraph with the same set of nodes for a subset of layers \( m \subset \{1, 2, \ldots, L\} \), where \(|m| > 1\).
Mathematically, \( C \) can be defined as
\[
C = \{(V_k^l, E_k^l) : V_k^l \subseteq V_l, E_k^l \subseteq V_k^l \times V_k^l, V_k^l = V_k^l, l, k \in m, m \subseteq \{1, 2, \ldots, L\}, |m| > 1\}.
\]

**Definition 2.** A private community \( C \) in a multiplex network \( \{G_l\} \) is defined as any community that is not common across at least two layers.

For a multiplex network with \( L \) layers and adjacency matrices, \( A_l \in \mathbb{R}^{n \times n}, l \in \{1, 2, \ldots, L\} \), we model each layer’s adjacency matrix in terms of common and individual communities using ONMTF. The resulting objective function can be formulated as
\[
\arg\min_{H \geq 0, S_l \geq 0, G_l \geq 0} \sum_{l=1}^{L} ||A_l - HS_l H^\top - H_l G_l H_l^\top||^2_F
\]
\[\text{s.t. } H H^\top = I, H_l H_l^\top = I, \text{ with } l \in \{1, 2, \ldots, L\}.
\] (2)

where \( H \in \mathbb{R}^{n \times k_c} \) and \( H_l \in \mathbb{R}^{n \times k_{pl}}, l \in \{1, 2, \ldots, L\} \) are the community membership matrices corresponding to the common and private communities, respectively, and \( S_l \) and \( G_l \) are diagonal matrices whose entries indicate the strength of the common and private communities across layers, respectively. In this work, it is assumed that the \( L \) layers have a total of \( k_c \) common communities and \( k_{pl} \) private communities in each layer \( l \). The goal is to simultaneously identify communities that are common across any subset of two or more layers and communities that are unique to each layer. Therefore, \( H \) will contain information for all common communities.

**B. OPTIMIZATION SOLUTION**

ONMTF optimization problem in (2) can be solved using a multiplicative update algorithm (MUA) [50]. Multiplicative update algorithms for solving NMF problems were introduced in [55], while solving NMTF with orthogonal constraints was first addressed by [50]. In this paper, we follow their approach to derive the multiplicative update rules for each variable.

To find the update rules for \( H, H_l, S_l \), and \( G_l \), the following Lagrangian function with Lagrange multipliers \( \Lambda \) and \( \Lambda_l \) is minimized:
\[
\mathcal{L}(H, H_l, S_l, G_l) = 
\sum_{l=1}^{L} ||A_l - HS_l H^\top - H_l G_l H_l^\top||^2_F
+ tr(\Lambda (H_l^\top H_l - I)) + \sum_{l=1}^{L} tr(\Lambda_l (H_l^\top H_l - I)).
\] (3)

For updating \( H \), we find \( \nabla_H \mathcal{L} \) as
\[
\nabla_H \mathcal{L} = \sum_{l=1}^{L} (4HS_l^\top H^\top S_l + 4H_l G_l^\top H_l^\top S_l - 4A_l S_l) + 4HA_l.
\] (4)

Setting \( \nabla_H \mathcal{L} = 0 \) and \( \nabla \mathcal{L} = 0 \), we obtain:
(i) \( \Lambda = \sum_{l=1}^{L} (-S_l^\top S_l - H_l^\top H_l G_l^\top H_l G_l^\top S_l + H_l^\top A_l S_l) \).
(ii) \( H_l^\top H_l = I_l \).

Substituting (i) and (ii) in Eq. (4), we get
\[
\nabla_H \mathcal{L} = \sum_{l=1}^{L} (4H_l G_l^\top H_l^\top S_l - 4A_l S_l + 4H_l^\top A_l S_l
- 4H_l G_l^\top H_l^\top S_l).
\] (5)

As discussed in [56], if the gradient of an error function, \( \epsilon \), is of the form \( \nabla \epsilon = \nabla \epsilon^+ - \nabla \epsilon^- \), where \( \nabla \epsilon^+ > 0 \) and \( \nabla \epsilon^- > 0 \), then the multiplicative update for parameter \( \Theta \) has the form \( \Theta = \Theta \odot \frac{\nabla \epsilon^+}{\nabla \epsilon^-} \). It can be easily seen that the multiplicative update preserves the nonnegativity of \( \Theta \), while \( \nabla \epsilon = 0 \) when the convergence is achieved. Following this procedure, from the gradient of the error function in Eq. (5), we derive the following multiplicative update rule for \( H_l \):
\[
H_l \leftarrow H_l \odot \frac{\sum_{l=1}^{L} (A_l S_l + 4H_l^\top A_l S_l + 4H_l G_l^\top H_l^\top S_l - 4H_l G_l^\top H_l^\top S_l)}{\sum_{l=1}^{L} (H_l G_l^\top H_l^\top S_l + 4H_l^\top A_l S_l)}.
\] (6)

This is the fixed point condition that any local minima \( H* \) must satisfy. This shows that if the update rule (6) converges, the converged solution is a local minimum of the optimization problem.

Similarly, we obtain the following update rules for \( H_l, S_l \), and \( G_l \), for each \( l \in \{1, 2, \ldots, L\} \):
\[
H_l \leftarrow H_l \odot \frac{A_l H_l G_l + H_l^\top H_l G_l^\top H_l G_l + H_l^\top G_l^\top A_l G_l^\top H_l^\top S_l}{H_l^\top H_l S_l + 4A_l S_l},
\] (8)
\[
S_l \leftarrow S_l \odot \frac{H_l^\top A_l H_l}{H_l^\top H_l S_l + 4A_l S_l},
\] (9)
\[
G_l \leftarrow G_l \odot \frac{H_l^\top A_l H_l}{H_l^\top H_l G_l^\top H_l G_l + H_l^\top G_l^\top A_l G_l^\top H_l^\top}. (10)
\]

Since NMF algorithms are initialized with random matrices, different runs yield local minima. For this reason, we run the algorithm 50 times and report the best results [57], [58]. As shown in Algorithm 1, for each random initialization of \( H, H_l, S_l \), and \( G_l \), the multiplicative update rules described in Eqs. (6)-(10) are repeated for 1000 iterations or until convergence. We then select the solution that yields the maximum value of the performance metric across the different runs. For synthetic networks for which a ground truth is available, Normalized Mutual Information (NMI) [59] is used. For networks without ground truth, Modularity Density \( Q_D \) [60] is used as the performance metric.
Algorithm 1 MX-ONMTF

**Input:** Adjacency matrices \( A_l, l \in \{1, 2, \ldots, L\} \)

**Output:** Community membership matrices \( H, H_l \)

1. Use Algorithm 2 to find \( k_c, k_l, k_p \).
2. for \( r = 1 \) to 50 do
3. Randomly initialize \( H, H_l, S_l, G_l > 0 \)
4. for 1000 iterations or until convergence do
5. update \( H \) according to Eq. (6)
6. update \( H_l \) for each \( l \in \{1, 2, \ldots, L\} \) according to Eq. (8)
7. update \( S_l \) for each \( l \in \{1, 2, \ldots, L\} \) according to Eq. (9)
8. update \( G_l \) for each \( l \in \{1, 2, \ldots, L\} \) according to Eq. (10)
9. end for
10. for each layer \( l \) do
11. Apply Algorithm 3 with \( A_l, H, \) and \( k_p \) as inputs to find \( H_l \).
12. for each \( i \) do
13. \( j^* = \arg\max_{j} \text{H}_{ij}(i,j) \)
14. if \( \text{H}_{ij}(i,j^*) > \text{H}_{ij}(i,j) \) then
15. \( \text{idx}(i) \leftarrow j^* \)
16. else
17. \( \text{idx}(i) \leftarrow (\arg\max_{j} \text{H}_{ij}(i,j)) + k_c + \sum_{n=m}^{l-1} k_p \)
18. end if
19. end for
20. Compute NMI, or \( Q_{D_l} \).
21. end for
22. Choose the partition \( r^* = \arg\max_r \text{NMI}_r \)

\( (r^* = \arg\max_r \text{NMI}_r) \).

C. NUMBER OF COMMUNITIES

In most NMF-based community detection algorithms, the number of communities \( k \) is an input parameter [48]. This problem is usually addressed by detecting communities with different values of \( k \) and selecting the one that gives the solution with the best pre-determined performance metric, such as modularity [61].

In this paper, a two-step approach is proposed to determine the number of communities per layer and the number of common communities. First, the number of communities per layer \((k_1, k_2, \ldots, k_L)\) are found using the eigengap rule [62] which determines the number of communities by the value that maximizes the eigengap, i.e. the difference between consecutive eigenvalues. A suitable null model, e.g., Laplacianized Erdős–Rényi adjacency matrices \( L^{null} \) with size and density matching the Laplacian of the network, can be used. The threshold, \( \delta \), can be set to be the 0.95 quantile of the largest eigengap (see lines 1 to 4 of Algorithm 2).

Next, ONMTF is applied to each layer [50] and the low-rank embedding matrices, \( U_l \in \mathbb{R}^{n \times k_l} \), are obtained. Once we have \( U_l \) corresponding to each layer \( l \), our goal is to reduce the embedding subspace by finding columns that are similar to each other, i.e., embedding vectors for the common communities. Each element of the embedding matrices, \( U_l(i,j) \), represents the likelihood of node \( i \) belonging to community \( j \). An agglomerative hierarchical clustering algorithm using Euclidean distance is applied to the columns of \( X = [U_1, U_2, \ldots, U_L] \in \mathbb{R}^{n \times m} \), where \( m = \sum_{l=1}^{L} k_l \), to obtain the number of common communities. At each step of the algorithm, the two columns with the smallest distance are aggregated, and the distances between the newly formed cluster and the remaining ones are updated. The assumption is that if two or more layers share a common community, the columns of the respective \( U_l \)’s corresponding to this community will be close to each other. A dendrogram like the one shown in Fig. 2 can be used to represent the different iterations of this algorithm. The \( m \) leaves of the dendrogram correspond to the total number of communities across the \( L \) layers.

This agglomerative hierarchical clustering algorithm outputs a matrix \( F \in \mathbb{R}^{(m-1) \times 3} \). The first two columns of \( F(i,:) \) correspond to the labels of the two leaves of the dendrogram that form cluster \( m + i \), and the third column contains the distance between these two leaves. The number of clusters resulting from this procedure corresponds to \( k_c \), while the columns of \( X \) corresponding to each layer \( l \) that are not assigned to any of the clusters correspond to \( k_{pl} \) (see lines 13 to 26 in Algorithm 2). The algorithm iterates until the minimum distance between any two clusters increases by more than 50% of the minimum distance from the previous iteration. Fig. 2 shows the dendrogram of the hierarchical clustering of the columns of the embedding matrices \( U_1, U_2, U_3 \) of a 3-layer network with \( k_1 = 6 \), \( k_2 = 6 \), and \( k_3 = 5 \) and the red line indicates where the
algorithm stops. For this example, \( k_c = 3, k_{p1} = 3, k_{p2} = 4, \) and \( k_{p3} = 3. \)

**D. DETERMINING THE COMMON COMMUNITY LABELS FOR EACH LAYER**

\( \mathbf{H} \in \mathbb{R}^{n \times k_c} \) is the community membership matrix corresponding to the common communities. In order to determine whether a node from a particular layer belongs to any of the \( k_c \) common communities, \( \mathbf{H} \) needs to go through some post-processing as described in Algorithm 3. First, for each node, \( i, \) in each layer, \( l, \) the common community membership matrix, \( \mathbf{H}, \) and the layer specific community membership matrix, \( \mathbf{H}_l, \) are concatenated and the column \( j \) with the maximum entry is identified (see line 3 in Algorithm 3). This determines the initial community assignment for that node. Next, we construct a binary common community membership matrix, \( \mathbf{Z}_l, \) for each layer where each entry is equal to 1 if a particular node belongs to one of the \( k_c \) common communities in that layer. For each layer, we compute the ratio of the average strength within a particular common community to the average strength outside that common community (lines 10-15). Finally, the common communities for that layer are determined as the ones which have the top \( k_l - k_{p_l} \) ratios (lines 16-18). As shown in Fig. 1c, the new embedding matrices corresponding to the common communities in each layer, \( \mathbf{H}_{c1}, \mathbf{H}_{c2}, \) and \( \mathbf{H}_{c3}, \) will only have the columns that contain the information corresponding to the common communities present in that layer. In this example, \( \mathbf{H}_{c1} \) keeps the two columns (purple and green communities) from \( \mathbf{H}, \) while \( \mathbf{H}_{c2} \) keeps only the first column (purple), and \( \mathbf{H}_{c3} \) only the second column (green).

**E. TIME COMPLEXITY**

The time complexity of the proposed algorithm is mostly due to the Multiplicative Updates Rules, Eqs. (6)-(10). The time complexity for the product of two matrices, e.g., the product of a \( m \times k \) matrix by a \( k \times n \) matrix, is \( O(mnk) \). Table 3 shows the time complexities of Eqs. (6)-(10) and the total complexity, with \( l \in \{1, 2, \ldots, L\}. \)

**F. STORAGE COMPLEXITY**

The storage complexity of our algorithm is determined by the sizes of the matrices \( \mathbf{H}, \mathbf{H}_l, \mathbf{S}_l, \) and \( \mathbf{G}_l. \) It can be seen that the total storage complexity is \( O(n(max(k, k_c), k_{p1}, \ldots, k_{pL})) \). For a multiplex network of size \( n \times n \times L, \) this is a significant reduction in memory cost.

**V. CONVERGENCE ANALYSIS**

In this section, we will prove the convergence of the multiplicative update rule defined by Eq. (6) using the auxiliary function approach. As the other update rules are similar, we will not explicitly prove their convergence. We first introduce the definition of auxiliary function as follows.

**Definition 3.** A function \( Z(\mathbf{H}, \mathbf{H}') \) is called an auxiliary function of \( \mathcal{L}(\mathbf{H}) \) if it satisfies

\[
Z(\mathbf{H}, \mathbf{H}') \geq \mathcal{L}(\mathbf{H}) \text{ and } Z(\mathbf{H}, \mathbf{H}) = \mathcal{L}(\mathbf{H}).
\]

The auxiliary function is a useful concept because of the following lemma which is proved in [55].

**Lemma 1.** If \( Z \) is an auxiliary function, then \( \mathcal{L} \) is non-increasing under the update.
TABLE 3. Storage complexity of each variable.

|        | \( \mathbb{O}(\max\{k_c, k_2, k_p\}) \) |
|--------|----------------------------------------|
| \( H \) | \( \mathbb{O}(n) \)                  |
| \( H_i \) | \( \mathbb{O}(n) \)                  |
| \( S_i \) | \( \mathbb{O}(n) \)                  |
| \( G_i \) | \( \mathbb{O}(n) \)                  |
| Total   | \( \mathbb{O}(n\max\{k_c, k_2, k_p\}) \) |

\[
\mathbf{H}^{t+1} = \underset{\mathbf{H}}{\text{argmin}} \ Z(\mathbf{H}, \mathbf{H}^t).
\]

**Theorem 1.** Given \( H_i, S_i, \) and \( G_i \) the Lagrangian function \( L(\mathbf{H}) \) is monotonically decreasing under the update rule (6).

**Proof:** For convenience, let \( L(h) \) denote the part of \( L(\mathbf{H}) \) dependent on \( H_i \). From Eq. (5) we have

\[
L'(h) = \sum_{l=1}^{L} (4H_l^{t+1}G_l^\top H_l^t H_l S_l - 4A_l^t H_l S_l) + 4HH^\top A_l S_l - 4HH^\top H_l G_l^\top H_l^\top H_l S_l).
\]

The second-order derivative of \( L(h) \) with respect to \( h_{ij} \) is

\[
L''(h) = \sum_{l=1}^{L} (4(H_l G_l H_l^\top)_{ij} S_l - 4A_l^t S_l) + 4([H_l^\top A_l H_l S_l]_{ij} + [H_l H_l^\top A_l^t S_l]) - 4([H_l^\top H_l G_l H_l^\top H_l S_l]_{ij} + [H_l^\top H_l G_l H_l^\top H_l S_l])_{ij}
\]

Let \( h'_{ij} \) denote the updated value of \( h_{ij} \) after the \( r \)th iteration, then the Taylor series expansion of \( L(h) \) at \( h'_{ij} \) can be written as

\[
L(h) = L(h'_{ij}) + L'(h'_{ij})(h - h')_{ij} + \frac{1}{2} L''(h'_{ij})(h - h')_{ij}^2.
\]

Now, the key is to find an appropriate auxiliary function \( Z(h, h'_{ij}) \). We choose the following \( Z(h, h'_{ij}) \) and prove in Appendix, that it satisfies the conditions to be an auxiliary function of \( L(h) \).

\[
Z(h, h'_{ij}) = L(h'_{ij}) + 3L'(h'_{ij})(h - h')_{ij} + \sum_{l=1}^{L} \left( 4H_l^{t+1}G_l^\top H_l^t H_l S_l + 4H_l^\top H_l G_l^\top A_l^t S_l \right)_{ij} \frac{h_{ij}}{2}(h - h')_{ij}^2.
\]

According to Lemma 1, we must find the minimum of \( Z(h, h'_{ij}) \) with respect to \( h \).

\[
\frac{\partial Z(h, h'_{ij})}{\partial h} = 3 \sum_{l=1}^{L} \left( 4H_l^{t+1}G_l^\top H_l^t H_l S_l \right)_{ij}(h - h')_{ij} + 3L'(h'_{ij}) + \sum_{l=1}^{L} \left( 4H_l^\top H_l^t A_l^t H_l S_l \right)_{ij}(h - h')_{ij} = 0
\]

Replacing \( L'(h'_{ij}) \) in the equation above and canceling the common terms, we obtain

\[
\sum_{l=1}^{L} (-4A_l^t H_l S_l - 4H_l^\top H_l G_l^\top H_l^t H_l S_l)_{ij} + \sum_{l=1}^{L} (4H_l^{t+1}G_l^\top H_l^t H_l S_l + 4H_l^\top H_l G_l^\top A_l^t H_l S_l)_{ij} \frac{h_{ij}}{2} = 0.
\]

Replacing \( h \) by \( h^{t+1}_{ij} \) we obtain the following update rule

\[
h_{ij}^{t+1} = \frac{h_{ij}^{t}}{\sum_{l=1}^{L} (4H_l^{t+1}G_l^\top H_l^t H_l S_l + 4H_l^\top H_l G_l^\top A_l^t H_l S_l)_{ij}},
\]

which is the same as the update rule shown in Eq. (6). 

**VI. EXPERIMENTS**

**A. SYNTHETIC MULTIPLEX NETWORKS**

1) MODEL DESCRIPTION

Multiplex benchmark networks based on the model described in [63] and [64] were generated. The authors in [63] propose a two-step approach to generate multilayer networks with a community structure. First, a multilayer partition with the user-defined number of nodes in each layer, number of layers, and an interlayer dependency tensor that specifies the desired dependency structure between layers is generated. Next, for the given multilayer partition, edges in each layer are generated following a degree-corrected block model [65] parameterized by the distribution of expected degrees and a community mixing parameter \( \mu \in [0, 1] \). The mixing parameter \( \mu \) controls the modularity of the network. When \( \mu = 0 \), all edges lie within communities, whereas \( \mu = 1 \) implies that edges are distributed independently. For multiplex networks, the probabilities in the interlayer dependency tensor are the same for all pairs of layers and are specified by \( p \in [0, 1] \). When \( p = 0 \), the partitions are independent across layers while \( p = 1 \) indicates an identical partition across layers.

In this paper, we extend the model described above to generate multiplex benchmark networks with common and private communities. We first generate the common communities by randomly selecting \( n_c \) nodes across all layers and setting the inter-layer dependency probability to \( p_1 \). For each common community, we decide whether it exists in a particular layer or not. Next, we independently generate the private communities for each layer with the remaining nodes in that layer. We generated 100 different random realizations of each multiplex network in order to report the average performance metric on the experiments.

2) EVALUATION

We compared the performance of our method to well-known multiplex community detection algorithms. In particular, we compared with ONMTF applied to the aggregated multiplex networks using the average of the adjacency matrices (Aggregated Average), Spectral Clustering on Multi-Layer graphs (SC-ML) [18], Generalized Louvain (GenLouvain) multilayer community detection algorithm [21], [66], Infomap [20], Collective Symmetric Nonnegative Matrix Factorization (CSNMF) [45], Collective Projective Nonnegative Matrix Factorization (CPNMF) [45], Collective Symmet-
ric Nonnegative Matrix Tri-factorization (CSNMTF) [45], and Orthogonal Link Matrix Factorization (OLMF) [30].

3) EXPERIMENT 1
In this experiment, we generated two different types of networks, one where the common communities are present across all layers and another where the common communities are present in different subsets of layers. Fig. 3 illustrates a single realization of the adjacency matrices generated with \( \mu = 0.1 \) for two 5-layer networks, one of each type (first row and second row). The network in Fig. 3a-3e has two common communities (the first two communities in each layer) across all layers and \( k_{p1} = 4, k_{p2} = 4, k_{p3} = 3, k_{p4} = 2, \) and \( k_{p5} = 2 \), while the network in Fig. 3f-3j has a total of 3 common communities (highlighted in red) that are present in different subsets of layers and \( k_{p1} = 3, k_{p2} = 4, k_{p3} = 3, k_{p4} = 3, \) and \( k_{p5} = 3 \). In order to evaluate the performance of our algorithm for different noise levels, these two types of networks were generated with varying values of \( \mu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\} \). The inter-layer dependency probability is \( p_{1} = 1 \), for the common communities.

Fig. 4 shows the results for the networks with 2 common communities across all layers for 3 (4a), 4 (4b), and 5 layers (4c), and for the networks with 3 common communities across different subsets of layers for 3 (4d), 4 (4e), and 5 layers (4f). The results indicate that our method performs well for both networks with common communities across all layers as well as for networks with common communities that do not span all layers. Our method discovers the complete structure of the network rather than forcing it to have a consensus partition. Moreover, our method is robust to noise for larger values of \( \mu \) compared to the other methods. We can also conclude that our algorithm performs better when the common communities are across all layers (see Fig. 4a, 4b, 4c) than when the multiplex community structure is more complex with common communities across subsets of layers (see Fig. 4d, 4e, 4f), but it still outperforms the rest of the methods. From Fig. 4 we can see that GenLouvain performs well when \( \mu \) is small, but its performance deteriorates for \( \mu \) values above 0.6. Another observation is that when the number of layers is small, the NMF-based methods perform closer to GenLouvain but when the number of layers increases, NMF algorithms perform worse. This is because these NMF methods perform aggregation on either the adjacency or the community indicator matrices. When there is more variation across layers, these methods fail to capture this heterogeneity.

4) EXPERIMENT 2
In the second experiment, we evaluated the robustness of the algorithm against variations in the common community structure by fixing \( \mu = 0.1 \) and varying the inter-layer dependency probability, \( p_{1} \), i.e., the common communities are allowed to vary across layers. The performance of all methods for a 5-layer network are reported in Fig. 5 based on the average NMI over 100 realizations of the network. As we can see in Fig. 5, our method still outperforms the other eight methods when there is some variation in the common community structure. This demonstrates that our method is robust to variations in the common community structure across layers. However, our algorithm is more sensitive to the drop in \( p_{1} \) than the rest of the methods. This is because when the common communities have a high variation our algorithm may try to assign some of those nodes to the private communities.

5) EXPERIMENT 3
Another parameter in our model is the number of common communities \( k_{c} \). In this experiment, we fixed \( n = 256, \mu = 0.3 \), and the number of communities in each layer, and varied \( k_{c} \) from 1 to 7. When \( k_{c} = 7 \), all communities are common across layers and there are no private communities. As we can see in Fig. 6, as \( k_{c} \) is increased, the performance of all the other methods improves, as expected, because these methods are designed to detect the common community structure. When the communities are common across layers, most of the methods, except...
Infomap, converge to the same NMI value. The performance of MX-ONMTF is not affected by increasing $k_c$, and when all communities are common it performs similarly to the other methods.

6) SCALABILITY ANALYSIS
In this experiment, we evaluate the effect of network size on the run time of the proposed algorithm. For this purpose, we fixed $\mu = 0.3$, $L = 5$, $k_c = 3$ and varied $n$ from 32 to 8192. From Fig. 7, it can be seen that our method’s run time is almost log-linear. This is comparable with all the other NMF based methods. However, our as shown in the previous experiments, our method performs better. Most of this time complexity is due to the multiplicative update rule used in NMF-based algorithms and can be reduced using alternative approaches as discussed in [67].

B. REAL WORLD MULTIPLEX NETWORKS
1) LAZEGA LAW FIRM MULTIPLEX SOCIAL NETWORK
Lazega Law Firm [68] is a multiplex social network with 71 nodes and three layers representing Co-work, Friendship and Advice relationships between partners and associates of a corporate law firm. This dataset also includes information
Applying MX-ONMTF to this network, we obtain one common community across all layers composed of the nodes colored in red as well as private communities for each layer, as shown in Fig. 8. This network does not have ground truth community structure, but we can compute the NMI between the detected community structure and each type of node attributes, i.e., metadata, to gain better insight into the results and to be able to provide quantitative results [24]. For each of the attributes, the nodes are divided into communities based on that particular attribute. For example, for the status, the network is divided into two communities, partners and associates. For Age and Seniority, the nodes were grouped into five-year bins. The community structure for each attribute is used as ground truth to compute the NMI between each attribute and the community structure detected by our method. The NMI values given in Table 4 for the partition obtained by our method, suggest that office location and type of practice (litigation or corporate) are highly correlated with community membership across co-work, friendship and advice relationships. We can also see that the partition detected by MX-ONMTF has greater NMI values for each attribute. Therefore, our method detects a community structure that takes all of the attributes into account instead of partitioning with respect to just one attribute as the Aggregated Average does.

2) C. ELEGANS NETWORK
C. Elegans Network [69], [70] is a multiplex network with 279 nodes and 3 layers representing different synaptic junctions (electric, chemical monadic, and polyadic) of 279 neurons of the Caenorhabditis Elegans connectome. Information about different attributes of the neurons in this dataset such as the type of neuron (motor neurons, sensory neurons, interneurons), and the color (blue, red, yellow, orange, etc.) is available.

Table 5 shows the NMI values between the community structures detected by each method and each of the three attributes available for this dataset. The partition detected by MX-ONMTF has greater NMI values for each of the attributes compared to the other eight methods.

3) YEASTLANDSCAPE MULTIPLEX NETWORK
Yeast Landscape is a multiplex genetic interaction network of a specie of yeast, Saccharomyces Cerevisiae [69], [71]. This network has 4458 nodes and 4 layers representing the positive and negative interaction networks of genes in Saccharomyces cerevisiae and positive and negative correlation based networks in which genes with similar interaction profiles are connected to each other. For this paper, we use the bioprocess annotations of the genes available on the supplementary data file S6 of [71] as ground truth communities.

| Method         | Status | Gender | Office | Seniority | Age | Practice | Law School |
|----------------|--------|--------|--------|-----------|-----|----------|------------|
| GenLouvain     | 0.0345 | 0.0307 | 0.3294 | 0.0807    | 0.0331 | 0.5468   | 0.0040     |
| Aggregated Average | 0.0383 | 0.0197 | 0.5379 | 0.1307    | 0.0798 | 0.4411   | 0.0201     |
| SC-ML          | 0.0138 | 0.0259 | 0.0731 | 0.1225    | 0.0484 | 0.0249   | 0.0040     |
| Infomap        | 0.0179 | 0.0043 | 0.1668 | 0.2880    | 0.0083 | 0.0003   | 0.0093     |
| CSNMF          | 0.0418 | 0.0291 | 0.5732 | 0.1155    | 0.0736 | 0.4227   | 0.0172     |
| CPNMF          | 0.0061 | 0.0524 | 0.1139 | 0.0514    | 0.0291 | 0.0187   | 0.0221     |
| CSNMTF         | 0.0395 | 0.0217 | 0.0798 | 0.0795    | 0.0487 | 0.1335   | 0.0279     |
| OLMF           | 0.0534 | 0.0296 | 0.4776 | 0.1023    | 0.0401 | 0.1665   | 0.0163     |
| MX-ONMTF       | 0.4752 | 0.4906 | 0.7386 | 0.4135    | 0.4203 | 0.6162   | 0.4226     |

| Method         | Neuron Group | Neuron Type | Color |
|----------------|--------------|-------------|-------|
| GenLouvain     | 0.3736       | 0.1297      | 0.2362|
| Aggregated Average | 0.3839      | 0.1590      | 0.2977|
| SC-ML          | 0.0185       | 0.0103      | 0.2690|
| Infomap        | 0.2205       | 0.2345      | 0.2355|
| CSNMF          | 0.1635       | 0.0757      | 0.1211|
| CPNMF          | 0.0854       | 0.0277      | 0.0628|
| CSNMTF         | 0.1113       | 0.0402      | 0.0914|
| OLMF           | 0.3288       | 0.2149      | 0.2669|
| MX-ONMTF       | 0.4074       | 0.4001      | 0.4593|
truth. We divided the genes into 18 groups according to their primary bioprocess. There were 1580 genes in this network without attributes.

Table 6 shows the NMI values between the community structures detected by each method and the bioprocess of the genes. MX-ONMTF gives the highest NMI value followed by the other NMF-based community detection methods.

### C. MULTIVIEW NETWORKS

In order to evaluate the performance of our method on networks where the communities are common across all layers, we use two multiview datasets, UCI Handwritten Digits\(^1\) [72] and Caltech [73].

The UCI Handwritten Digits dataset consists of features of handwritten digits from 0-9 extracted from a collection of Dutch utility maps. There is a total of 2000 patterns that have been digitized in binary images, 200 patterns per digit. These digits are represented by six different feature sets: Fourier coefficients of the character shapes, profile correlations, Karhunen-Loève coefficients, pixel averages in 2 × 3 windows, Zernike moments, and morphological features. Each layer of the multiplex network represents one of the 6 features. The graphs are constructed using \(k\)-nearest neighbors graphs with the nearest 50 neighbors and Euclidean distance.

Caltech-101 is a well-known object recognition dataset that consists of pictures of objects belonging to 102 categories. There are about 40 to 800 images per category for a total of 9144 images. This dataset consists of 6 types of features extracted from each image. A multiplex network with 6 layers representing each of the features, 102 classes, and 9144 nodes is constructed from this dataset using \(k\)-nearest neighbors graphs with the nearest 50 nodes. A smaller version of this dataset is also used in these experiments, where only 20 objects are selected, resulting in a multiplex network with 6 layers, 20 communities, and 2386 nodes.

In this case, as we have the true class assignment, we compute the NMI with respect to this ground truth. As it can be seen in Table 7, our method performs better than the rest of the methods for the three networks. This indicates that even in cases where there are no private communities, our method is successful at obtaining the consensus community structure, thus can be used as an alternative to multiview clustering.

\(^{1}\)https://archive.ics.uci.edu/ml/datasets/Multiple+Features

### VII. CONCLUSION

In this paper, we proposed a multiplex community detection method based on ONMTF. The proposed method, MX-ONMTF, is able to detect both common and private communities across layers, allowing us to differentiate between the topologies across layers. The proposed algorithm is based on multiplicative update rules, and a proof of convergence is provided. A new approach based on the eigengap criterion and agglomerative hierarchical clustering is introduced for determining the number of communities. Results for both synthetic and real-world networks show that our method performs better than existing community detection methods for multiplex networks as it is able to handle the heterogeneity of the network topology across layers. Moreover, experiments on multiview networks show that our method also performs well in cases where a consensus community structure is needed.

### APPENDIX. AUXILIARY FUNCTION PROOF

**Proposition 1.** The following function, \(Z(h, h'_l)\),

\[
Z(h, h'_l) = \mathcal{L}(h'_l) + 3\mathcal{L}(h'_l)(h - h'_l) + \frac{3}{2} \sum_{l=1}^{L} (4H_lG_lH_l^\top H_l^\prime S_l + 4H_l^\top A_lH_l^\prime S_l)_{h'_l}(h - h'_l)^2
\]

is an auxiliary function of \(\mathcal{L}(H)\).

**Proof:** First, when \(h = h'_l\), the equality \(Z(h, h'_l) = \mathcal{L}(h)\) holds. Now, we need to show that \(Z(h, h'_l) \geq \mathcal{L}(h)\).

It can be seen that the first and second terms of \(Z(h, h'_l)\) are greater than the first and second terms in \(\mathcal{L}(h)\), Therefore, it suffices to show that \(\frac{3}{2} \sum_{l=1}^{L} (4H_lG_lH_l^\top H_l^\prime S_l + 4H_l^\top A_lH_l^\prime S_l)_{h'_l} \geq \mathcal{L}'(h'_l)\).

It can be shown that

\[
\begin{align*}
(H_lG_lH_l^\top H_l^\prime S_l)_{h'_l} &= \sum_{p,q} (H_lG_lH_l^\top)_{pq} h'_l^p h'_l^q \\
&\geq (H_lG_lH_l^\top)_{pq} S_l^p S_l^q,
\end{align*}
\]

\[
\begin{align*}
(H_l^\top A_lH_l^\prime S_l)_{h'_l} &= \sum_{p,q} h'_l^p (H_l^\top A_lH_l^\prime S_l)_{pq} \\
&\geq (H_l^\top A_lH_l^\prime S_l)_{h'_l},
\end{align*}
\]

\[1\]
\[
\begin{align*}
(HH^\top A_hS)_ij & = \sum_p q h_{ij}^p h_{pj}^q (A_hS)_ij \\
H^\top H A_hS & \geq \sum_m m^j h_{ij}^j (A_hS)_ij \\
H^\top A_hS & \geq (HH^\top A_hS)_ij.
\end{align*}
\]

Therefore,
\[
3 \frac{(HH^\top A_hS)_ij}{H^\top A_hS} \geq (H^\top A_hS)_ij + h_{ij}^j (A_hS)_ij + (H^\top A_hS)_ij,
\]
and thus
\[
\sum_j \left(\sum_l (H_hG_hH^\top A_hS + HH^\top A_hS)_{jl} h_{jl} \right) \geq L''(h_{ij}^j).
\]

Therefore, (11) is an auxiliary function of \(L(h)\).

REFERENCES

[1] A.-L. Barabási, “Network science,” Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci., vol. 371, no. 1987, 2013, Art. no. 20120375.

[2] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, “Multilayer networks,” J. Complex Netw., vol. 2, no. 3, pp. 203–271, Sep. 2014.

[3] S. E. Smith-Aguilar, F. Aureli, L. Busia, C. Schaffner, and G. Ramos-Fernández, “Using multiplex networks to capture the multidimensional nature of social structure,” Primates, vol. 60, no. 3, pp. 277–295, May 2019.

[4] A. Aleta, S. Meloni, and Y. Moreno, “A multilayer perspective for the analysis of urban transportation systems,” Sci. Rep., vol. 7, no. 1, pp. 1–9, Mar. 2017.

[5] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” Phys. Rep., vol. 659, pp. 1–44, Nov. 2016.

[6] Y. Li, K. He, K. Kloster, D. Bindel, and J. Hopcroft, “Local spectral clustering for overlapping community detection,” ACM Trans. Knowl. Discovery Data, vol. 12, no. 2, pp. 1–27, Apr. 2018.

[7] Y. van Gennip, B. Hunter, R. Ahn, P. Elliott, K. Luh, M. Halvorson, S. Reid, M. Valaski, J. Wo, G. E. Tita, A. L. Bertozzi, and P. J. Brantingham, “Community detection using spectral clustering on sparse geosocial data,” SIAM J. Appl. Math., vol. 73, no. 1, pp. 67–83, Jan. 2013.

[8] E. Abe, “Community detection and stochastic block models: Recent developments,” J. Mach. Learn. Res., vol. 18, no. 1, pp. 6446–6531, Jan. 2017.

[9] M. Chen, K. Kuzmin, and B. K. Szymanski, “Community detection via heterogeneous multilayer community detection,” in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), Aug. 2015, pp. 1308–1315.

[10] M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall, “Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems,” Phys. Rev. X, vol. 5, no. 1, Mar. 2015, Art. no. 011027.

[11] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, “Community structure in time-dependent, multiscale, and multiplex networks,” Science, vol. 328, no. 5980, pp. 876–878, May 2010.

[12] A. Amelio and C. Pizzuti, “Community detection in multidimensional networks,” in Proc. IEEE 26th Int. Conf. Tools With Artif. Intell., Nov. 2014, pp. 352–359.

[13] G. Zhu and K. Li, “A unified model for community detection of multiplex networks,” in Proc. Int. Conf. Web Inf. Syst. Eng. Cham, Switzerland: Springer, 2014, pp. 31–46.

[14] A. R. Panfil, S. D. Howison, R. Lambiotte, and M. A. Porter, “Relating modularity maximization and stochastic block models in multiplex networks,” SIAM J. Math. Data Sci., vol. 1, no. 4, pp. 667–698, Jan. 2019.

[15] S. Paul and Y. Chen, “Spectral and matrix factorization methods for consistent community detection in multi-layer networks,” Ann. Statist., vol. 48, no. 1, pp. 230–250, 2020.

[16] S. Bhattacharyya and S. Chatterjee, “Spectral clustering for multiple sparse networks,” J. Mach. Learn. Res., vol. 18, no. 1, pp. 1308–1315, May 2017.

[17] Z. Kuncheva and G. Montana, “Community detection in multiplex networks using locally adaptive random walks,” in Proc. 9th IEEE Int. Conf. Data Mining, Nov. 2009, pp. 1016–1021.

[18] Y. Jing, T. Li, Z. Lu, and D. Xia, “Community detection on mixture networks via regularized tensor decomposition,” Ann. Statist., vol. 49, no. 6, pp. 3181–3205, Dec. 2021.

[19] Q. Han, K. S. Xu, and E. M. Airolı, “Consistent estimation of dynamic and multi-layer block models,” 2014, arXiv:1410.8597.

[20] J. Lei, K. Chen, and B. Lynch, “Consistent community detection in multi-layer network data,” Biomетrika, vol. 107, no. 1, pp. 61–73, Mar. 2020.

[21] G. Braun, H. Tyagi, and C. Biermacki, “Clustering multilayer graphs with missing nodes,” in Proc. Int. Conf. Artif. Intell. Statist., 2021, pp. 2260–2268.

[22] W. Tang, Z. Lu, and I. S. Dhillon, “Clustering with multiple graphs,” in Proc. IEEE Conf. Data Mining, Dec. 2009, pp. 1016–1021.

[23] B.-Y. Jiang, B. Li, Z. Ye, and D. Xia, “Community detection on mixture networks via regularized tensor decomposition,” Ann. Statist., vol. 49, no. 6, pp. 3181–3205, Dec. 2021.

[24] X. Fan, M. Pensky, F. Yu, and T. Zhang, “ALMA: Alternating minimization algorithm for clustering multilayer network,” J. Mach. Learn. Res., vol. 23, no. 1, pp. 14855–14900, 2022.

[25] F. Tang, X. Zhao, and C. Li, “Community detection in multilayer networks based on matrix factorization and spectral embedding method,” Mathematics, vol. 11, no. 7, p. 1573, Mar. 2023.

[26] N. Stanley, S. Shai, D. Taylor, and P. J. Mucha, “Clustering network layers with the strata multilayer stochastic block model,” IEEE Trans. Netw. Sci. Eng., vol. 3, no. 2, pp. 95–105, Apr. 2016.

[27] H. T. Ali, S. Liu, Y. Yilmaz, R. Couillet, I. Rajapakse, and A. Hero, “Latent heterogeneous multilayer community detection,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2019, pp. 8142–8146.

[28] C. De Bacco, E. A. Power, D. B. Larremore, and C. Moore, “Community detection, link prediction, and layer interdependence in multilayer networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 95, no. 4, Apr. 2017, Art. no. 042317.

[29] O. Boutemine and M. Bouguessa, “Mining community structures in multidimensional networks,” ACM Trans. Knowl. Discovery From Data, vol. 11, no. 4, pp. 1–36, Nov. 2017.

[30] F. Karimi, S. Lotfi, and H. Izadkhah, “Multilayer community detection in complex networks using an evolutionary approach,” Expert Syst. Appl., vol. 146, May 2020, Art. no. 113184.

[31] C. Chen, M. K. Ng, and S. Zhang, “Block spectral clustering methods for multiple graphs,” Numer. Linear Algebra With Appl., vol. 24, no. 1, p. e2075, Jan. 2017.
[40] C. Ding, X. He, and H. D. Simon, “On the equivalence of nonnegative matrix factorization and spectral clustering,” in Proc. SIAM Int. Conf. Data Mining, Apr. 2005, pp. 606–610.

[41] H. Wang, F. Nie, H. Huang, and C. Ding, “Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation,” in Proc. IEEE 11th Int. Conf. Data Mining, Dec. 2011, pp. 774–783.

[42] S. Mankad and G. Michailidis, “Structural and functional discovery in dynamic networks with non-negative matrix factorization,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Fluids Relat. Interdiscip. Top., vol. 88, no. 4, Oct. 2013, Art. no. 042812.

[43] B.-J. Sun, H. Shen, J. Gao, W. Ouyang, and X. Cheng, “A non-negative symmetric encoder–decoder approach for community detection,” in Proc. ACM Conf. Inf. Knowl. Manage., Nov. 2017, pp. 597–606.

[44] X. Ma, D. Dong, and Q. Wang, “Community detection in multi-layer networks using joint non-negative matrix factorization,” IEEE Trans. Knowl. Data Eng., vol. 31, no. 2, pp. 273–286, Feb. 2019.

[45] V. Gligorijevic, Y. Panagakis, and S. Zafeiriou, “Non-negative matrix factorizations for multiplex network analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 4, pp. 928–940, Apr. 2019.

[46] H. T. Nguyen, T. N. Dinh, and T. Vu, “Community detection in multiplex social networks,” in Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS), Apr. 2015, pp. 654–659.

[47] E. Cozzo, M. Kivelä, M. D. Domenico, A. Solé-Ribalta, A. Arenas, S. Gómez, M. A. Porter, and Y. Moreno, “Structure of triadic relations in multiplex networks,” New J. Phys., vol. 17, no. 7, Jul. 2015, Art. no. 073029.

[48] C. He, X. Fei, Q. Cheng, H. Li, Z. Hu, and Y. Tang, “A survey of community detection in complex networks using nonnegative matrix factorization,” IEEE Trans. Comput. Social Syst., vol. 3, no. 2, pp. 440–457, Apr. 2022.

[49] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, “Overlapping community detection using Bayesian non-negative matrix factorization,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Fluids Relat. Interdiscip. Top., vol. 83, no. 6, Jun. 2011, Art. no. 066114.

[50] C. Ding, T. Li, W. Peng, and H. Park, “Orthogonal nonnegative matrix t-factorizations for clustering,” in Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2006, pp. 126–135.

[51] F. Pompili, N. Gillis, P.-A. Absil, and F. Glineur, “Two algorithms for orthogonal nonnegative matrix factorization with application to clustering,” Neurocomputing, vol. 141, pp. 15–25, Oct. 2014.

[52] S. Wang, T.-H. Chang, Y. Cui, and J.-S. Pang, “Clustering by orthogonal NMF model and non-convex penalty optimization,” IEEE Trans. Signal Process., vol. 69, pp. 5273–5288, 2021.

[53] W. Wu, S. Kwong, Y. Zhou, Y. Jia, and W. Gao, “Nonnegative matrix factorization with mixed hypergraph regularization for community detection,” Inf. Sci., vol. 435, pp. 263–281, Apr. 2018.

[54] H. Lu, X. Sanga, Q. Zhao, and J. Lu, “Community detection algorithm based on nonnegative matrix factorization and pairwise constraints,” Phys. A. Stat. Mech. Appl., vol. 545, May 2020, Art. no. 123491.

[55] D. Leu and H. Seung, “Algorithms for non-negative matrix factorization,” in Proc. Adv. Neural Inf. Process. Syst., vol. 13, 2001, pp. 535–541.

[56] J. Yoo and S. Choi, “Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on Stiefel manifolds,” Inf. Process. Manage., vol. 46, no. 5, pp. 559–570, Sep. 2010.

[57] T. Li and C.-C. Ding, “Nonnegative matrix factorizations for clustering: A survey,” in Data Clustering, London, U.K.: Chapman & Hall, 2012, pp. 21–176.

[58] X. Luo, Z. Liu, L. Jin, Y. Zhou, and M. Zou, “Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 3, pp. 1203–1215, Mar. 2022.

[59] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, “Comparing community structure identification,” J. Stat. Mech. Theory Exp., vol. 2005, no. 9, p. P09008, Sep. 2005.

[60] J. Xu, J. C. Zeng, Y. W. Xue, and Y. Wang, “Quantitative function for community detection,” Adv. Mater. Res., vol. 433–440, pp. 6441–6446, Jan. 2012.

[61] S. Pramanik, R. Tackx, A. Navekall, J.-L. Guillaume, and B. Mitra, “Discovering community structure in multilayer networks,” in Proc. IEEE Int. Conf. Data Sci. Adv. (DASIA), Oct. 2017, pp. 611–620.

[62] F. Liu, D. Choi, L. Xie, and K. Roeder, “Global spectral clustering in dynamic networks,” Proc. Nat. Acad. Sci. USA, vol. 115, no. 5, pp. 927–932, Jan. 2018.

[63] M. Bazzi, L. G. S. Jeub, A. Arenas, S. D. Howison, and M. A. Porter, “A framework for the construction of generative models for mesoscale structure in multilayer networks,” 2016, p. 20, arXiv:1608.06196.

[64] L. Jeub and M. Bazzi. (2016). A Generative Model for Mesoscale Structure in Multilayer Networks Implemented in MATLAB. [Online]. Available: https://github.com/MultilayerGM/MultilayerGM-MATLAB.

[65] B. Karrer and M. E. J. Newman, “Stochastic blockmodels and community structure in networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Fluids Relat. Interdiscip. Top., vol. 83, no. 1, Jan. 2011, Art. no. 011917.

[66] I. S. Jutla, L. G. Jeub, and P. J. Mucha, “A generalized Louvain method for community detection implemented in MATLAB,” 2011. [Online]. Available: http://netwiki.amath.unc.edu/GenLouvain.

[67] E. Lazega, “The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership,” London, U.K.: Oxford Univ. Press, 2001.

[68] M. De Domenico, M. A. Porter, and A. Arenas, “MuxViz: A tool for multilayer analysis and visualization of networks,” J. Complex Netw., vol. 3, no. 2, pp. 159–176, Jun. 2015.

[69] B. L. Chen, D. H. Hall, and D. B. Chklovskii, “Wiring optimization can relate neuronal structure and function,” Proc. Nat. Acad. Sci. USA, vol. 103, no. 12, pp. 4723–4728, Mar. 2006.

[70] M. Costanzo, A. Baryshnikova, J. Bellay, Y. Kim, E. D. Spear, C. S. Sevier, H. Ding, J. L. Koh, K. Toffighi, S. Mostafavi, and J. Prinz, “The genetic landscape of a cell,” Science, vol. 327, no. 5964, pp. 425–431, Jan. 2010.

[71] D. Dutta and D. Graff, (2017). UCI Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml.

[72] F. F. Li, M. Andreetto, M. Ranzato, and P. Perona, “Caltech 101 (1.0) [data set].” CaltechDATA,” Tech. Rep., 2022, doi: 10.22022/D1.20086.

MEIBY ORTIZ-BOUZA (Graduate Student Member, IEEE) received the B.S. degree in telecommunications and electronics engineering from the Technological University of Havana, Cuba, in 2017, and the M.S. degree in applied mathematics and scientific computation from Bowling Green State University, OH, USA, in 2020. She is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, Michigan State University, East Lansing, MI, USA. Her research interests include graph signal processing and machine learning.

SELIN AVIYENTE (Senior Member, IEEE) received the B.S. degree (Honors) in electrical and electronics engineering from Boğaziçi University, Istanbul, Turkey, in 1997, and the M.S. and Ph.D. degrees in electrical engineering: systems and scientific computation from Bowling Green State University, OH, USA, in 1999 and 2002, respectively. She joined the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA, in 2002, where she is currently a Professor. Her research interests include statistical signal processing, higher order data representations, and complex network analysis. She was a recipient of the 2005 Withrow Teaching Excellence Award and the 2008 National Science Foundation (NSF) CAREER Award. She currently serves on several technical committees for IEEE Signal Processing Society. She is also an Associate Editor of IEEE OPEN JOURNAL OF SIGNAL PROCESSING.