Non-Negative Matrix Factorizations for Multiplex Network Analysis

Vladimir Gligorijević, Yannis Panagakis, and Stefanos Zafeiriou, *Member, IEEE*

Abstract—Networks have been a general tool for representing, analyzing, and modeling relational data arising in several domains. One of the most important aspect of network analysis is community detection or network clustering. Until recently, the major focus have been on discovering community structure in single (i.e., monoplex) networks. However, with the advent of relational data with multiple modalities, multiplex networks, i.e., networks composed of multiple layers representing different aspects of relations, have emerged. Consequently, community detection in multiplex network, i.e., detecting clusters of nodes shared by all layers, has become a new challenge. In this paper, we propose **N**etwork **F**usion for **C**omposite **C**ommunity **E**xtraction (**NF-CCE**), a new class of algorithms, based on four different non–negative matrix factorization models, capable of extracting composite communities in multiplex networks. Each algorithm works in two steps: first, it finds a non–negative, low–dimensional feature representation of each network layer; then, it fuses the feature representation of layers into a common non–negative, low–dimensional feature representation via collective factorization. The composite clusters are extracted from the common feature representation. We demonstrate the superior performance of our algorithms over the *state–of–the–art* methods on various types of multiplex networks, including biological, social, economic, citation, phone communication, and brain multiplex networks.

Index Terms—Multiplex networks, non-negative matrix factorization, community detection, network integration

1 Introduction

Networks (or graphs[1]) along with their theoretical foundations are powerful mathematical tools for representing, modeling, and analyzing complex systems arising in several scientific disciplines including sociology, biology, physics, and engineering among others [1]. Concretely, social networks, economic networks, biological networks, telecommunications networks, etc. are just a few examples of graphs in which a large set of entities (or agents) correspond to *nodes* (or *vertices*) and relationships or interactions between entities correspond to *edges* (or *links*). Structural analysis of these networks have yielded important findings in the corresponding fields [2], [3].

Community detection (also known as *graph clustering* or *module detection*) is one of the foremost problems in network analysis. It aims to find groups of nodes (i.e., *clusters, modules* or *communities*) that are more densely connected to each other than they are to the rest of the network [4]. Even thought the volume of research on community detection is large, e.g., [5], [6], [7], [8], [9], [10], [11], [12], [13], the majority of these methods focus on networks with only one type of relations between nodes (i.e., networks of single type interaction).

However, many real-world systems are naturally represented with multiple types of relationships, or with relationships that chance in time. Such systems include subsystems or layers of connectivity representing different modes of complexity. For instance, in social systems, users in social networks engage in different types of interactions (e.g., personal, professional, social, etc.). In biology, different experiments or measurements can provide different types of interactions between genes. Reducing these networks to a single type interactions by disregarding their multiple modalities is often a very crude approximation that fails to capture a rich and holistic complexity of the system. In order to encompass a multimodal nature of these relations, a *multiplex network* representation has been proposed [14], [15], [16]. Multiplex networks (also known as *multidimensional*, *multiview* or *multilayer* networks) have recently attracted a lot of attention in network science community. We re They can be represented as a set of graph layers that share a common set of nodes, but different set of edges in each layer (cf. Fig. 1). With the emergence of this network representation, finding composite communities across different layers of multiplex network has become a new challenge [17]. We refer a reader to Kivelä *et al.* [18] for a recent review on multiplex networks.

Here, distinct from the previous approaches (reviewed in Section 2), we focus on *multiplex community detection*. Concretely, a novel and general model, namely the **N**etwork **F**usion for **C**omposite **C**ommunity **E**xtraction (NF-CCE) along with its algorithmic framework is developed in Section 3. The heart of the NF-CCE is the Collective Non-negative Matrix Factorization (CNMF), which is employed in order to collectively factorize adjacency matrices representing different layers in the network. The collective factorization facilitate us to obtain a consensus low-dimensional latent representation, shared across the decomposition, and hence to reveal the communities shared between the net-

- *Y. Panagakis and S. Zafeiriou are with the Department of Computing, Imperial College London, UK.*
- *Y. Panagakis is also with the Department of Computer Science, Middlesex University London, UK.*
- *V. Gligorijević was with the Department of Computing, Imperial College London, UK. He is now a Research Fellow at Flatiron Institute, Simons Foundation, USA. Corresponding author: V. Gligorijević, email: vgligorijevic@flatironinstitute.org*

1. we use terms *graphs* and *network* interchangeably throughout this paper

work layers. The contributions of the paper are as follows:

1) Inspired by recent advances in non-negative matrix factorization (NMF) techniques for graph clustering [19], [20] and by using tools for subspace analysis on Grassmann manifold [21], [22], [23], we propose a general framework for extracting composite communities from multiplex networks. In particular, the framework NF-CCE, utilizes four different NMF techniques, each of which is generalized for collective factorization of adjacency matrices representing network layers, and for computing a consensus low-dimensional latent feature matrix shared across the decomposition that is used for extracting latent communities common to all network layers. To this end, a general model involving factorization of networks' adjacency matrices and fusion of their low-dimensional subspace representation on Grassmann manifold is proposed in Section 3.
2) Unlike a few matrix factorization-based methods for multiplex community extraction that have been proposed so far, e.g., [23], [24], [25], [26], that directly decompose matrices representing network layers into a low-dimensional representation common to all network layers, NF-CCE is conceptually different. Namely, it works in two steps: first, it denoises each network layer by computing its non-negative low-dimensional representation. Then it merges the low-dimensional representations into a consensus low-dimensional representation common to all network layers. This makes our method more robust to noise, and consequently, it yields much stable clustering results.
3) Four efficient algorithms based on four different NMF techniques are developed for NF-CCE using the concept of *natural gradient* [22], [23] and presented in the form of *multiplicative update rules* [27] in Section 3.

The advantages of the NF-CCE over the state-of-the-art in community detection are demonstrated by conducting extensive experiments on a wide range of real-world multiplex networks, including biological, social, economic, citation, phone communication, and brain multiplex networks. In particular, we compared the clustering performance of our four methods with 6 state-of-the-art methods and 5 baseline methods (i.e., single-layer methods modified for multiplex networks), on 9 different real-world multiplex networks including 3 large-scale multiplex biological networks of 3 different species. Experimental results, in Section 5, indicate that the proposed methods yield much stable clustering results than the *state-of-the-art* and baseline methods, by robustly handling incomplete and noisy network layers. The experiments conducted on multiplex biological networks of 3 different species indicate NF-CCE as a superior method for finding composite communities (in biological networks also known as *functional modules*). Moreover, the results also indicate that NF-CCE can extract unique and more functionally consistent modules by considering all network layers together than by considering each network layer separately.

Notations: throughout the paper, matrices are denoted by uppercase boldface letters, e.g., $\mathbf{X}$. Subscript indices denote matrix elements, e.g., $\mathbf{X}_{ij}$, whereas superscript indices in brackets denote network layer, e.g., $\mathbf{X}^{(i)}$. The set of real numbers is denoted by $\mathbb{R}$. $|\cdot|$ denotes the cardinality of a set, e.g., $|S|$. A binary matrix of size $n \times m$ is represented by $\{0,1\}^{n\times m}$.

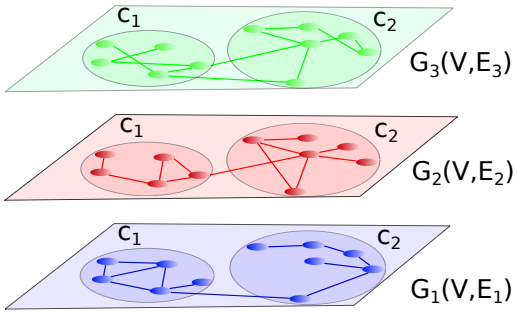

Fig. 1. An example of a multiplex network with 11 nodes present in three complementary layers denoted in different colors. Two different communities across all three layers can be identified.

2 Background and Related Work

2.1 Single-layer (monoplex) networks

In graph theory, a monoplex network (graph) can be represented as an ordered pair, $G = (V, E)$, where V is a set of $n = |V|$ vertices or nodes, and E is a set of $m = |E|$ edges or links between the vertices [28]. An adjacency matrix representing a graph G is denoted by $\mathbf{A} \in \{0,1\}^{n\times n}$, where $\mathbf{A}_{ij} = 1$ if there is an edge between vertices i and j, and $\mathbf{A}_{ij} = 0$ otherwise. Most of the real-world networks that we consider throughout the paper are represented as *edge-weighted graphs*, $G = (V, E, w)$, where $w : E \to \mathbb{R}$ assigns real values to edges. In this case, the adjacency matrix instead of being a binary matrix, is a real one i.e., $\mathbf{A} \in \mathbb{R}^{n\times n}$, with entries characterizing the strength of association or interaction between the network nodes.

Although, there is no universally accepted mathematical definition of the community notion in graphs, the probably most commonly accepted definition is the following: a community is a set of nodes in a network that are connected more densely among each other than they are to the rest of the network [4], [29]. Hence, the problem of community detection is as follows: given an adjacency matrix $\mathbf{A}$ of one network with n nodes and k communities, find the community assignment of all nodes, denoted by $\mathbf{H} \in \{0,1\}^{n\times k}$, where $\mathbf{H}_{ij} = 1$ if nodes i belongs to community j, and $\mathbf{H}_{ij} = 0$ otherwise. We consider the case of *non-overlapping* communities, where a node can belong to only one community, i.e., $\sum_{j=1}^{k} \mathbf{H}_{ij} = 1$.

To address the community detection problem in monoplex networks, several methods have been proposed. Comprehensive surveys of these methods are [4] and [13]. To make the paper self-contained, here, we briefly review some of the most representative approaches, including *graph partitioning, spectral clustering, hierarchical clustering, modularity maximization, statistical inference and structure-based methods*, as well as method that rely on *non-negative matrix factorizations*:

- *Graph partitioning* aims to group nodes into partitions such that the *cut size*, i.e., the total number of edges between any two partitions, is minimal. Two widely used graph partitioning algorithms that also take into account the size of partitions are *Ratio Cut* and *Normalized Cut* [30]. Graph partitioning algorithms can be alternatively defined as spectral algorithms in which the objective is to partition the nodes into communities based on their eigenvectors obtained from *eigendecomposition of graph Laplacian matrix* [31].
- In *hierarchical clustering* the goal is to reveal network communities and their hierarchical structure based on a similarity (usually topological) measure computed between pairs of nodes [32].
- *Modularity-based* algorithms are among the most popular ones. Modularity was designed to measure the strength of partition of a network into communities. It is defined as a fraction of the edges that fall within the community minus the expected fraction when these edges are randomly distributed [5], [29]. Various algorithms have been proposed for modularity optimization, including greedy techniques, simulated annealing, spectral optimization, etc. [4].
- *Statistical inference* methods aims at fitting the *generative model* to the network data based on some hypothesis. Most commonly used statistical inference method for community detection is the *stochastic block model*, that aims to approximate a given adjacency matrix by a block structure [33]. Each block in the model represents a community.
- *Structure-based* methods aim to find subgraphs representing meta definitions of communities. Their objective is to find maximal cliques, i.e., the cliques which are not the subgraph of any other clique. The union of these cliques form a subgraph, whose components are interpreted as communities [34].
- More recently, graph clustering methods that rely on the *Non-Negative Matrix Factorization* (NMF) [27] have been proposed e.g., [19], [20]. Their goal is to approximate a symmetric adjacency matrix of a given network by a product of two non-negative, low-rank matrices, such that they have clustering interpretation, i.e., they can be used for assigning nodes to communities. The proposed methods here, follow this line of research, but as opposed to the existing methods [19], [20], the NF-CCE can effectively handle multiplex networks.

2.2 Multiplex networks

A multiplex network is a set of N monoplex networks (or layers), $G_i(V, E_i)$, for $i = 1, \ldots, N$. The number of nodes in each layer is the same, $n = |V|$, while the connectivity pattern and the distribution of links in each layer differs, $m_i = |E_i|$ (see Fig. 1). Similarly to monoplex networks, we consider the case where each layer represents a weighted, undirected graph, i.e., $G_i(V, E_i, w_i)$. A multiplex network can be represented as a set of adjacency matrices encoding connectivity patterns of individual layers, $\mathbf{A}^{(i)} \in \mathbb{R}^{n \times n}$, for $i = 1, \ldots, N$. The goal of community detection in multiplex networks is to infer shared, latent community assignment that best fits all given layers. Given that each layer contains incomplete and complementary information, this process of finding shared communities by integrating information from all layers is also known in the literature as *network integration (fusion)* [25], [35].

Unlike the case of monoplex networks, research on community detection in multiplex networks is scarce. Existing methods extract communities from multiplex networks first by aggregating the links of all layers into a single layer, and then applying a monoplex method to that single layer [25], [36], [37]. However, this approach does not account for shared information between layers and treats the noise present in each layer uniformly. Clearly, this is not the case in real-world multiplex networks where each level is contaminated by different noise in terms of magnitude and, possibly, distribution. Thus, by aggregating links from different layers the noise in the aggregated layer significantly increases, resulting in a poor community detection performance.

Current state-of-the-art methods are built on monoplex approaches and further generalized to multiplex networks. They can be divided into the following categories:

- *Modularity-based* approaches that generalize the notion of modularity from single-layer to multi-layer networks. Namely, to alleviate the above mentioned limitations, the Principal Modularity Maximization (PMM) [25] has been proposed. First, for each layer, PMM extracts structural features by optimizing its modularity, and thus significantly denoising each layer; then, it applies PCA on concatenated matrix of structural feature matrices, to find the principal vectors, followed by K-means to perform clustering assignment. The main drawback of this approach is that it treats structural feature matrices of all layers on equal basis (i.e., it is not capable of distinguishing between more and less informative network layers, or complementary layers). Even though the noise is properly handled by this method, the *complementarity aspect* cannot be captured well by the integration step. On the other hand, Mucha *et al.* [16] proposed a modularity-based method for multiplex and also time-dependent networks that can handle layer complementarity. Namely, in addition to inter-layer connections, their generalized modularity function takes into account the coupling between nodes of different layers. To optimize the modularity function, the authors proposed *Generalized Louvain (or GenLouvain, GL)* algorithm that has been shown to be most efficient algorithm for their generalized modularity function [38]. Despite the efficiency, the method is not designed to find consensus clustering assignment across different layers; instead, it only provides node clustering assignment for each individual layer.
- *Spectral clustering* approaches that generalize the eigendecomposition from single to multiple Laplacian matrices representing network layers. One of the *state-of-the-art* spectral clustering methods for multiplex graphs is the Spectral Clustering on Multi-Layer (SC-ML) [21]. First, for each network layer, SC-ML computes a subspace spanned by the principal eigen-

vectors of its Laplacian matrix. Then, by interpreting each subspace as a point on Grassmann manifold, SC-ML merges subspaces into a consensus subspace from which the composite clusters are extracted. The biggest drawback of this methods is the underlying spectral clustering, that always finds tight and small-scale and, in some cases, almost trivial communities. For example, SC-ML cannot adequately handle network layers with missing or weak connections, or layers that have disconnected parts.

- *Information diffusion-based* approaches that utilize the concept of diffusion on networks to integrate network layers. One of such methods is Similarity Network Fusion (SNF) proposed by Wang *et al.* [39]. SNF captures both shared and complementary information in network layers. It computes a fused matrix from the similarity matrices derived from all layers through parallel interchanging diffusion process on network layers. Then, by applying a spectral clustering method on the fused matrix they extract communities. Another widely used method that uses the concept of dynamics on network (i.e., diffusion) is Multiplex Infomap [40]. Namely, Multiplex Infomap optimizes the map equation, which exploits the information-theoretic duality between network dimensionality reduction, and the problem of network community detection. However, for noisy networks, the diffusion process, i.e., information propagation, is not very efficient and it may results in poor clustering performance [41].
- *Matrix and tensor factorization-based* approaches that utilize collective factorization of adjacency matrices representing network layers. A few matrix and tensor decomposition-based approaches have been proposed so far [24], [26], [42], [43], [44]. Tang *et al.* [42] introduced the Linked Matrix Factorization (LMF) which fuses information from multiple network layers by factorizing each adjacency matrix into a layer-specific factor and a factor that is common to all network layers. Dong *et al.* [26], introduced the Spectral Clustering with Generalized Eigendecomposition (SC-GED) which factorizes Laplacian matrices instead of adjacency matrices. Papalexakis *et al.* [43] proposed GraphFuse, a method for clustering multi-layer networks based on sparse PARAllel FACtor (PARAFAC) decomposition [45] with non-negativity constraints. A similar approach has been adopted by Gauvin *et al.* [44] who used PARAFAC decomposition for time-varying networks. Cheng *et al.* introduced Co-regularized Graph Clustering based on NMF (CGC-NMF). They factorize each adjacency matrix using symmetric NMF while keeping the Euclidean distance between their non-negative low-dimensional representations small. As already pointed out in Section 1, one of the major limitations of all of these factorization methods is that they treat each network layer on an equal basis and, unlike PMM or SC-ML, for example, they cannot filter out irrelevant information or noise.

To alleviate the drawbacks of the aforementioned methods, the NF-CCE framework is detailed in the following section. It consists of 4 models, where each layer is first denoised by computing its non-negative low-dimensional subspace representation. Then, the low-dimensional subspaces are merged into a consensus subspace whose non-negative property enables clustering interpretation. The models are conceptually similar to the SC-ML method, since they use the same merging technique to find the common subspace of all layers.

3 Proposed Framework

Here, we present four novel metods that are built upon 4 non-negative matrix factorization models, SNMF [19], PNMF [46], SNMTF [47] and Semi-NMTF [48], and extended for fusion and clustering of multiplex networks. Since the derivation of each method is similar, we present them in a unified framework, namely NF-CCE. NF-CCE extracts composite communities from a multiplex network consisting of N layers. In particular, given N-layered multiplex network represented by adjacency matrices, $\{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}\}$, NF-CCE consists of two steps:

Step 1: For each network layer, i, we obtain its non-negative, low-dimensional representation, $\mathbf{H}^{(i)}$, under column orthonormality constraints i.e., $\mathbf{H}^{(i)T}\mathbf{H}^{(i)} = \mathbf{I}$, by using any of the non-negative factorization methods mentioned above.

Step 2: We fuse the low-dimensional representations into a common, consensus representation, $\mathbf{H}$, by proposing a collective matrix factorization model. That is, we collectively decompose all adjacency matrices, $\mathbf{A}^{(i)}$ into a common matrix, $\mathbf{H}$, whilst enforcing the non-negative low-dimensional representation of network layers, $\mathbf{H}^{(i)}$ (computed in the previous step), to be close enough to the consensus low-dimensional representation, $\mathbf{H}$. The general objective function capturing these two properties is written as follows:

$$\min_{\mathbf{H}\geq 0} \mathcal{J} = \sum_{i=1}^{N} \mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) + \alpha \sum_{i=1}^{N} \mathcal{J}_c^{(i)}(\mathbf{H}; \mathbf{H}^{(i)}) \qquad (1)$$

where, $\mathcal{J}^{(i)}$ is an objective function for clustering ith layer and $\mathcal{J}_c^{(i)}$ is the loss function quantifying the inconsistency between each low-dimensional representation $\mathbf{H}^{(i)}$, computed in *Step 1*, and the consensus representation $\mathbf{H}$.

Below we provide the details of the second step for each individual factorization technique.

3.1 Collective SNMF (CSNMF)

We factorize each individual adjacency matrix using Symmetric NMF in the following way:

$$\mathbf{A}^{(i)} \approx \mathbf{H}\mathbf{H}^T$$

under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{H}^T\mathbf{H} = \mathbf{I}$; where, $i = 1, \ldots, N$.

The first part of our general objective function in *Step 2* (Eq. 1) has the following form:

$$\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) = \| \mathbf{A}^{(i)} - \mathbf{H}\mathbf{H}^T \|_F^2 \qquad (2)$$

where $\mathbf{H}$ is the consensus, non-negative low-dimensional matrix, and F denotes Frobenius norm.

3.2 Collective PNMF (CPNMF)

We factorize each individual adjacency matrix using Projective NMF in the following way:

$$\mathbf{A}^{(i)} \approx \mathbf{H}\mathbf{H}^T\mathbf{A}^{(i)}$$

under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{H}^T\mathbf{H} = \mathbf{I}$; where, $i = 1, \ldots, N$.

The first part of our general objective function in *Step 2* (Eq. 1) has the following form:

$$\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) = \| \mathbf{A}^{(i)} - \mathbf{H}\mathbf{H}^T\mathbf{A}^{(i)} \|_F^2 \quad (3)$$

where $\mathbf{H}$ is the consensus, non-negative low-dimensional matrix.

3.3 Collective SNMTF (CSNMTF)

We tri-factorize each individual adjacency matrix using Symmetric NMTF in the following way:

$$\mathbf{A}^{(i)} \approx \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T$$

under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{H}^T\mathbf{H} = \mathbf{I}$; where $i = 1, \ldots, N$.

The first part of our general objective function in *Step 2* (Eq. 1) has the following form:

$$\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}, \mathbf{S}^{(i)}) = \| \mathbf{A}^{(i)} - \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T \|_F^2 \quad (4)$$

where $\mathbf{H}$ is the consensus low-dimensional matrix.

In the derivation of our algorithm, we distinguish between two cases. In the first case, we consider $\mathbf{S}$ matrix to be non-negative, i.e., $\mathbf{S}^{(i)} \geq 0$. We call that case CSNMTF. In the second case, we consider $\mathbf{S}^{(i)}$ matrix to have both positive and negative entries. We call this case collective symmetric semi-NMTF, or CSsemi-NMTF (or CSsNMTF).

3.4 Merging low-dimensional representation of graph layers on Grassmann Manifolds

For the second term in our general objective function (Eq. 1) in *Step 2*, we utilize the orthonormal property of non-negative, low-dimensional matrices, $\mathbf{H}^{(i)}$, and propose a distance measure based on this property. Namely, Dong *et al.* [21] proposed to use the tools from subspace analysis on Grassmann manifold. A Grassmann manifold $\mathbb{G}(k, n)$ is a set of k-dimensional linear subspaces in $\mathbb{R}^n$ [21]. Given that, each orthonormal cluster indicator matrix, $\mathbf{H}_i \in \mathbb{R}^{n \times k}$, spanning the corresponding k-dimensional non-negative subspace, $span(\mathbf{H}_i)$ in $\mathbb{R}^n$, is mapped to a unique point on the Grassmann manifold $\mathbb{G}(k, n)$. The geodesic distance between two subspaces, can be computed by projection distance. For example, the square distance between two subspaces, $\mathbf{H}_i$ and $\mathbf{H}_j$, can be computed as follows:

$$\begin{aligned} d^2_{proj}(\mathbf{H}_i, \mathbf{H}_j) &= \sum_{i=1}^{k} sin^2\theta_i = k - \sum_{i=1}^{k} cos^2\theta_i \\ &= k - tr(\mathbf{H}_i\mathbf{H}_i^T\mathbf{H}_j\mathbf{H}_j^T) \end{aligned}$$

where, $\{\theta_i\}_{i=1}^k$ are principal angles between k-dimensional subspaces, $span(\mathbf{H}_i)$ and $span(\mathbf{H}_j)$.

To find a consensus subspace, $\mathbf{H}$, we factorize all the adjacency matrices, $\mathbf{A}^{(i)}$, and minimize the distance between their subspaces and the consensus subspace on Grassmann manifold. Following this approach, we can write the second part of our general objective function in the following way:

$$\begin{aligned} \mathcal{J}_c^{(i)}(\mathbf{H}, \mathbf{H}^{(i)}) &= k - tr(\mathbf{H}\mathbf{H}^T\mathbf{H}^{(i)}\mathbf{H}^{(i)T}) \\ &= \| \mathbf{H}\mathbf{H}^T - \mathbf{H}^{(i)}\mathbf{H}^{(i)T} \|_F^2 \end{aligned} \quad (5)$$

3.5 Derivation of the general multiplicative update rule

In *Step 1*, we use well-known non-negative factorization techniques, namely SNMF, PNMF, SNMTF and Ssemi-NMTF, for which the update rules for computing low-dimensional non-negative matrices, $\mathbf{H}^{(i)}$, have been provided in the corresponding papers [19], [46], [47], [48], respectively. They are summarized in Table 1. As for the *Step 2*, we derive the update rules for each of the collective factorization techniques presented in Sections 3.1, 3.2 and 3.3. The details of the derivation are given in Section 2 of the online supplementary material. Here we provide a general update rule for Equation 1.

We minimize the general objective function shown in Equation 1, under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{H}^T\mathbf{H} = \mathbf{I}$. Namely, we adopt the idea from Ding *et al.* [47] to impose orthonormality constraint on $\mathbf{H}$ matrix, i.e., $\mathbf{H}^T\mathbf{H} = \mathbf{I}$; that has been shown to lead to a more rigorous clustering interpretation [47]. Moreover, assignments of network nodes to composite communities can readily be done by examining the entries in rows of $\mathbf{H}$ matrix. Namely, we can interpret matrix $\mathbf{H}^{n \times k}$ as the *cluster indicator matrix*, where the entries in i-th row (after row normalization) can be interpreted as a posterior probability that a node i belongs to each of the k composite communities. In all our experiments, we apply *hard clustering* procedure, where a node is assign to the cluster that has the largest probability value.

We derive the update rule, for matrix $\mathbf{H}$, for minimizing the objective function (Eq. 1) following the procedure from the constrained optimization theory [49]. Specifically, we follow the strategy employed in the derivation of NMF [27] to obtain a multiplicative update rule for $\mathbf{H}$ matrix that can be used for finding a local minimum of the optimization problem (Eq. 1).

The derivative of the objective function (Eq. 1) with respect to $\mathbf{H}$ is as follows:

$$\nabla_{\mathbf{H}}\mathcal{J} = \sum_{i=1}^{N} \nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) - \alpha \sum_{i=1}^{N} \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H} \quad (6)$$

where the first term under summation can be decomposed into two non-negative terms, namely:

$$\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) = [\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)})]^+ - [\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)})]^-$$

where, $[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)})]^+ \geq 0$, $[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)})]^- \geq 0$ are non-negative terms. Depending on the type of collective factorization technique represented in Section 3.1, 3.2 or 3.3, the first term represents the derivative of the corresponding objective function, i.e., Equation 2, 3 or 4, respectively. The

second term represents a derivative of Equation 5 with respect to $\mathbf{H}$.

To incorporate the orthonormality constraint into the update rule, we introduce the concept of *natural gradient* by following the work of Panagakis *et al.* [23]. Namely, we shown in Section 3.4 that columns of $\mathbf{H}$ matrix span a vector subspace known as Grassmann manifold $\mathbb{G}(k,n)$, i.e., $span(\mathbf{H}) \in \mathbb{G}(k,n)$ [23]. Using that, Amari in [22] has showed that when an optimization problem is defined over a Grassmann manifold, the ordinary gradient of the optimization function (Equation 6) does not represent its steepest direction, but *natural gradient* does [22].

Therefore, we define a natural gradient to optimize our objective function (1) under the orthornormality constraint. Following Panagakis *et al.* [23], the natural gradient of $\mathcal{J}$ on Grassmann manifold at $\mathbf{H}$ can be written in terms of the ordinary gradient as follows:

$$\widetilde{\nabla}_{\mathbf{H}}\mathcal{J} = \nabla_{\mathbf{H}}\mathcal{J} - \mathbf{H}\mathbf{H}^T\nabla_{\mathbf{H}}\mathcal{J} \tag{7}$$

where, $\nabla_{\mathbf{H}}\mathcal{J}$ is the ordinary gradient given in Equation 6.

Following the Karush-Kuhn-Tucker (KKT) complementarity condition [49] and preserving the non-negativity of $\mathbf{H}$, the general update rule for $\mathbf{H}$ matrix using the natural gradient is as follows:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk}\frac{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^-_{jk}}{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^+_{jk}} \tag{8}$$

$$[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^- = \mathbf{H}\mathbf{H}^T\sum_{i=1}^{N}[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^+ + \sum_{i=1}^{N}[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^- + \alpha\sum_{i=1}^{N}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}$$

$$[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^+ = \mathbf{H}\mathbf{H}^T\sum_{i=1}^{N}[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^- + \sum_{i=1}^{N}[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^+ + \alpha\mathbf{H}\mathbf{H}^T\sum_{i=1}^{N}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}$$

The concrete update rule for each collective factorization method is summarized in Table 1 and united within *NF-CCE* algorithm (see Algorithm 1).

Algorithm 1 NF-CCE

Input: Adjacency matrices $\mathbf{A}^{(i)}$ for each network layer $i = 1,\ldots,N$; number of clusters k; parameter α; factorization technique: FACTORIZATION $\in$ {SNMF, PNMF, SNMTF}

Output: Consensus cluster indicator matrix $\mathbf{H}$

switch (FACTORIZATION)
case 'SNMF':
  for $i \in [1,N]$ **do**
    $\mathbf{H}^{(i)} \leftarrow$ FACTORIZATION$(\mathbf{A}^{(i)},k)$
  end for
  $\mathbf{A}_{avg} \leftarrow \sum_{i=1}^{N}\mathbf{A}^{(i)} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}$
  $\mathbf{H} \leftarrow$ FACTORIZATION$(\mathbf{A}_{avg},k)$
case 'PNMF':
  for $i \in [1,N]$ **do**
    $\mathbf{H}^{(i)} \leftarrow$ FACTORIZATION$(\mathbf{A}^{(i)},k)$
  end for
  $\mathbf{A}_{avg} \leftarrow \sum_{i=1}^{N}\mathbf{A}^{(i)}\mathbf{A}^{(i)T} + \alpha\mathbf{H}^{(i)}\mathbf{H}^{(i)T}$
  $\mathbf{H} \leftarrow$ FACTORIZATION$(\mathbf{A}_{avg},k)$
case 'SNMTF':
  for $i \in [1,N]$ **do**
    $(\mathbf{H}^{(i)},\mathbf{S}^{(i)}) \leftarrow$ FACTORIZATION$(\mathbf{A}^{(i)},k)$
  end for
  $\mathbf{H} \leftarrow$ CSNMTF$(\{\mathbf{A}^{(i)}\}_{i=1}^{N},\{\mathbf{H}^{(i)}\}_{i=1}^{N},\{\mathbf{S}^{(i)}\}_{i=1}^{N},k)$
end switch

4 EXPERIMENTS

We test our methods on synthetic, as well as on real-world data. We designed synthetic multiplex networks with clear ground truth information and different properties in terms of noise and complementary information of network layers. The goal is to address the robustness of our methods against noise and their ability to handle complementary information contained in layers. The real-world multiplex networks are taken from diverse experimental studies to demonstrate the applicability of our methods in a broad spectrum of disciplines. Namely, we consider social and biological networks, networks of mobile phone communications, brain networks and networks constructed from bibliographic data. The biological networks are treated as a special case because of their lack of ground truth clusters. We provide detailed analysis of such networks based on the functional annotations of their nodes (genes). We present results of comparative analysis of our proposed methods against *state-of-the-art* methods described in Section 2.2. Specifically, we compare our methods against PMM, SC-ML, SNF, LMF, GraphFuse and CGC-NMF. Moreover, we adopt the following single-layer methods, SNMF, SNMTF, PNMF and MM (modularity maximization) to be our baseline methods.

4.1 Synthetic multiplex networks

We generate two sets of synthetic multiplex networks. First type, that we denote *SYNTH-C*, is designed to demonstrate complementary information in layers; whereas the second type, that we denote *SYNTH-N*, is designed to demonstrate different levels of noise between communities contained in layers. Our synthetic networks are generated by using *planted partition model* [50]. The procedure is as follows: we choose the total number of nodes n partitioned into N communities of equal or different sizes. For each layer, we split the corresponding adjacency matrix into blocks defined by the partition. Entries in each diagonal block, are filled with ones randomly, with probability p_{ii}, representing the *within-community probability* or also referred as community edge density. We also add random noise between each pair of blocks, ij, with probability p_{ij}, representing *between-community probability*. The larger the values of p_{ij} are the harder the clustering is. Similarly, the smaller the values of p_{ii} are the harder the clustering is. We vary these probabilities across the layers to simulate complementary information and noise in the following way:

SYNTH-C. We generate two-layer multiplex networks with $n = 200$ nodes and $N = 2$ communities with equal number of nodes each. We generate 11 different multiplex networks with different amounts of information between two layers. Namely, we vary the within-community probability $p_{11} = \{0.05, 0.075, \ldots, 0.3\}$ of the first community

TABLE 1
Multiplicative update rules (MUR) for single-layer and multiplex network analysis.

Method	Single-layer MUR	Multiplex MUR
SNMF	$\mathbf{H}^{(i)}_{jk} \leftarrow \mathbf{H}^{(i)}_{jk} \frac{\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\right]_{jk}}$	$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\left[\mathbf{A}_{avg}\mathbf{H}\right]_{jk}}{\left[\mathbf{H}\mathbf{H}^{T}\mathbf{A}_{avg}\mathbf{H}\right]_{jk}}$ $\mathbf{A}_{avg} = \sum_{i=1}^{N} \mathbf{A}^{(i)} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}$
PNMF	$\mathbf{H}^{(i)}_{jk} \leftarrow \mathbf{H}^{(i)}_{jk} \frac{\left[\mathbf{A}^{(i)}\mathbf{A}^{(i)T}\mathbf{H}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{A}^{(i)T}\mathbf{H}^{(i)}\right]_{jk}}$	$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\left[\mathbf{A}_{avg}\mathbf{H}\right]_{jk}}{\left[\mathbf{H}\mathbf{H}^{T}\mathbf{A}_{avg}\mathbf{H}\right]_{jk}}$ $\mathbf{A}_{avg} = \sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{A}^{(i)T} + \alpha\mathbf{H}^{(i)}\mathbf{H}^{(i)T}$
SNMTF	$\mathbf{H}^{(i)}_{jk} \leftarrow \mathbf{H}^{(i)}_{jk} \frac{\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]_{jk}}$ $\mathbf{S}^{(i)}_{jk} \leftarrow \mathbf{S}^{(i)}_{jk} \frac{\left[\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\right]_{jk}}$	$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\left[\sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)} + \frac{\alpha}{2}\sum_{i=1}^{N} \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\right]_{jk}}{\left[\mathbf{H}\mathbf{H}^{T}\left(\sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)} + \frac{\alpha}{2}\sum_{i=1}^{N} \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\right)\right]_{jk}}$
SsNMTF	$\mathbf{H}^{(i)}_{jk} \leftarrow \mathbf{H}^{(i)}_{jk} \frac{\left[\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]^{+} + \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]^{-}\right]_{jk}}{\left[\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]^{-} + \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]^{+}\right]_{jk}}$ $\mathbf{S}^{(i)} \leftarrow \left(\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\right)^{-1}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\left(\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\right)^{-1}$	$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\left[\sum_{i=1}^{N}\left[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\right]^{+} + \mathbf{H}\mathbf{H}^{T}\left[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\right]^{-} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\right]_{jk}}{\left[\sum_{i=1}^{N}\left[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\right]^{-} + \mathbf{H}\mathbf{H}^{T}\left(\left[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\right]^{+} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\right)\right]_{jk}}$

of the first layer across different multiplex networks, while fixing the within-community probability of the second community, $p_{22} = 0.2$. In the second layer, we represent the complementary information by fixing the within-community probability of the first community to $p_{11} = 0.2$ and varying within-cluster probability of the second community $p_{22} = \{0.05, 0.075, \ldots, 0.3\}$ across the multiplex networks. For all multiplex networks, we set the same amount of noisy links, by fixing between-community probability to $p_{12} = 0.05$.

SYNTH-N. Similar to *SYNTH-C* we generate two-layer multiplex networks with two communities (n and N are the same as in *SYNTH-C*). We fix the within-community probability of both communities and both layers to $p_{11} = 0.3$ and $p_{22} = 0.3$ across all multiplex networks. We vary the between-community probability $p_{12} = \{0.02, 0.04, \ldots, 0.2\}$ of the first layer, while keeping the between-community probability of the second layer fixed, $p_{12} = 0.02$, across all multiplex networks.

For each set of *within-community*, p_{ii}, and *between-community*, p_{ij}, probabilities, we generate 100 different random realizations of multiplex networks. This allows us to compute the average performance and the standard error for each artificial multiplex networks. The spy plots of adjacency matrices representing layers of *SYNTH-C* and *SYNTH-N* are given in the Section 1 of the online supplementary material.

4.2 Real-world multiplex networks

Below we provide a brief description of real-world multiplex networks used in our comparative study:

Bibliographic data, CiteSeer: the data are adopted from [51]. The network consist of $N = 3,312$ papers belonging to 6 different research categories, that we grouped into $k = 3$ pairs categories. We consider these categories as the ground truth classes. We construct two layers: citation layer, representing the citation relations between papers extracted from the paper citation records; and the paper similarity layer, constructed by extracting a vector of $3,703$ most frequent and unique words for each paper, and then computing the cosine similarity between each pair of papers. We construct the k-nearest neighbor graph from the similarity matrix by connecting each paper with its 10 most similar papers.

Bibliographic data, CoRA: the data are adopted from [51]. The network consists of $1,662$ machine learning papers grouped into $k = 3$ different research categories. Namely, Genetic Algorithms, Neural Networks and Probabilistic Methods. We use the same approach as for *CiteSeer* dataset to construct the citation and similarity layers.

Mobile phone data (MPD): the data are adopted from [26]. The network consists of $N = 3$ layers representing different mobile phone communications between $n = 87$ phone users on the MIT campus; namely, the layers represent physical location, bluetooth scans and phone calls. The ground truth clusters are known and manually annotated.

Social network data (SND): the data are adopted from [52]. The dataset represents the multiplex social network of a corporate law partnership, consisting of $N = 3$ layers having three types of edges, namely, co-work, friendship and advice. Each layer has $n = 71$ nodes representing employees in a law firm. Nodes have many attributes. We use the location of employees' offices as well as their status in the firm as the ground truth for clusters and perform two different experiments, namely *SND(o)* and *SND(s)* respectively.

Worm Brain Networks (WBN): the data are retrieved from WormAtlas[2], i.e., from the original study of White *et al.* [53]. The network consist of $n = 279$ nodes representing neurons, connected via $N = 5$ different types of links (i.e., layers), representing 5 different types of synapse. We use neuron types as the ground truth clusters.

Word Trade Networks (WTN): the data represents different types of trade relationships (import/export) among $n = 183$ countries in the world [54]. The network consist of 339 layers representing different products (goods). Since, for some products layers are very sparse, we retain the layers having more than $n - 1$ links, which resulted in $N =$ layers.

2. http://www.wormatlas.org/

TABLE 2
Real-world multiplex networks used for our comparative study.

Net name	n	N	ground truth	Ref.
CiteSeer	3,312	2	known ($k = 3$)	[51]
CoRA	1,662	2	known ($k = 3$)	[51]
MPD	87	3	known ($k = 6$)	[26]
SND	71	3	known ($k = 3$)	[52]
WBN	279	10	known ($k = 10$)	[53]
WTN	183	14	known ($k = 5$)	[54]
HBN	13,251	8	unknown ($k = 100$)	[55]
YBN	3,904	44	unknown ($k = 100$)	[55]
MBN	21,603	10	unknown ($k = 100$)	[55]

We use geographic regions (continents) of countries and economic trade categories for defining ground truth clusters[3]. Thus, we perform experiments with this two ground truth clusters, namely *WTN (reg)*, denoting geographic regions and *WTN (cat)*, denoting economic categories.

In Table 2 we summarize the important statistics and information of real-world multiplex networks used in our experiments.

4.2.1 Multiplex biological networks

We obtained multiplex biological networks of 3 different species, i.e., human biological network (HBN), yeast biological network (YBN) and mouse biological network (MBN)[4], from the study of *Mostafavi and Morris* [55]. The network layers are constructed from the data obtained from different experimental studies and from the publicly available databases. The network layers represent different types of interactions between genes[5], including protein interactions, genetic interaction, gene co-expressions, protein localization, disease associations, etc. The number of nodes and layers in each network is summarized in Table 2. For each network and its genes, the corresponding GO annotations has also been provided by Mostafavi and Morris [55]. For details about network statistics and layer representation we refer a reader to *Mostafavi and Morris* [55].

4.3 Setup for state-of-the-art methods

Each of the *state-of-the-art* method takes as an input parameter the number of clusters k that needs to be known in advance. Also, some of the methods take as input other types of parameters that needs to be determined. To make the comparison fair, below we briefly explain each of the comparing method and provide the implementation and parameter fitting details that we use in all our experiments (for detailed procedure on parameter fitting, please refer to Section 3 in the online supplementary material):

Baseline, single-layer methods (MM, SNMF, PNMF, SNMTF and SsNMTF). In order to apply them on multiplex network we first merge all the network layers into a single network described by the following adjacency matrix: $\mathbf{A} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A}^{(i)}$.

3. data about countries are downloaded from http://unctadstat.unctad.org

4. The network can be retrieved from: http://morrislab.med.utoronto.ca/~sara/SW/

5. genes and their coded proteins are considered as the same type of nodes in networks layers

PMM [25] has a single parameter, ℓ, which represents the number of structural features to be extracted from each network layer. In all our runs, we compare the clustering performance by varying this parameter, but we also noted that the clustering performance does not change significantly when $l \gg k$.

SNF [39] the method is parameter-free. However, the method prefers data in the kernel matrix. Thus, we use diffusion kernel matrix representation of binary interaction networks as an input to this method.

SC-ML [21] has a single regularization parameter, α, that balances the trade-off between two terms in the SC-ML objective function. In all our experiments we choose the value of α that leads to the best clustering performance.

LMF [42] has a regularization parameter, α, that balances the influence of regularization term added to objective function to improve numerical stability and avoid over fitting. We vary α in all our runs, and choose the value of α that leads to the best clustering performance.

GraphFuse [43] has a single parameter, sparsity penalty factor λ, that is chosen by exhaustive grid search and the value of λ that leads to the best clustering performance is chosen.

CGC-NMF [24] has a set of parameters $\gamma_{ij} \geq 0$ that balance between single-domain and cross-domain clustering objective for each pair of layers ij. Given that in all our experiments the relationship between node labels for any pair of layers is *one-to-one*, we set $\gamma_{ij} = 1$ (as in [24]) for all pairs of layers and throughout all our experiments.

GL [16] has two parameters: the resolution parameter, γ_s, that affects the size and number of communities in each layer, and the coupling parameter, ω, that affects the coupling strength between network layers. We chose the optimal parameters by exhaustive grid search.

Infomap [40] has one parameter, relax rate, r, which is a probability of diffusion process happening between layers. It is a measure of coupling between different layers. We use the top cluster indices for each node as a final clustering assignment, and we choose the value of r that leads to the best clustering performance.

4.4 Clustering evaluation measures

Here we discuss the evaluation measures used in our experiments to evaluate and compare the performance of our proposed methods with the above described *state-of-the-art* methods. Given that we test our methods on multiplex network with known and unknown ground truth cluster, we distinguish between two sets of measures:

Known ground truth. For multiplex network with known ground truth clustering assignment, we use the following three widely used clustering accuracy measures: *Purity* [56], *Normalized Mutual Information (NMI)* [57] and *Adjusted Rand Index (ARI)* [57]. All three measures provide a quantitative way to compare the computed clusters $\Omega = \{\omega_1, \ldots, \omega_k\}$ with respect to the ground truth classes: $C = \{c_1, \ldots, c_k\}$. *Purity* represents percentage of the total number of nodes classified correctly, and it is defined as [56]:

$$Purity(\Omega, C) = \frac{1}{n}\sum_{k}\max_{j}|\omega_k \cap c_j|$$

where n is the total number of nodes, and $|\omega_k \cap c_j|$ represents the number of nodes in the intersection of ω_k and c_j. To trade-off the quality of the clustering against the number of clusters we use *NMI*. NMI is defined as [57]:

$$NMI(\Omega, C) = \frac{I(\Omega; C)}{|H(\Omega) + H(C)|/2}$$

where I is the mutual information between node clusters Ω and classes C, while $H(\Omega)$ and $H(C)$ represent the entropy of clusters and classes respectively. Finally, *Rand Index* represents percentage of true positive (TP) and true negative (TN) decisions assigns that are correct (i.e., accuracy). It is defined as:

$$RI(\Omega, C) = \frac{TP + TN}{TP + FP + FN + TN}$$

where, FP and FN represent false positive and false negative decisions respectively. *ARI* is defined to be scaled in range $[0, 1]$ [57]. All three measures are in the range $[0, 1]$, and the higher their value, the better clustering quality is.

Unknown ground truth. For biological networks, the ground truth clusters are unknown and evaluating the clustering results becomes more challenging. In order to evaluate the functional modules identified by our methods, we use Gene Ontology (GO) [58], a commonly used gene annotation database. GO represents a systematic classification of all known protein functions organized as well-defined terms (also known as GO terms) divided into three main categories, namely Molecular Function (MF), Biological Process (BP) and Cellular Component (CC) [58]. GO terms (i.e., annotations), representing gene functions, are hierarchically structured where low-level (general) terms annotate more proteins than high-level (specific) terms. Thus, in our analysis we aim to evaluate our clusters with high-level (specific) GO terms annotating not more than 100 genes. Additionally, we remove GO terms annotating 2 or less proteins. Thus, for each gene in a network, we create a list of its corresponding GO term annotations. We then analyze the consistency of each cluster, i, obtain by our method, by computing the *redundancy* [6], R_i as follows:

$$R_i = 1 - \frac{\Big(- \sum_{l=1}^{N_{GO}} p_l \log_2 p_l \Big)}{\log_2 N_{GO}}$$

where, N_{GO} represents the total number of GO terms considered and p_l represents the relative frequency of GO term l in cluster i. Redundancy is based on normalized Shannon's entropy and its values range between 0 and 1. For clusters in which the majority of genes share the same GO terms (annotations) redundancy is close to 1, whereas for clusters in which the majority of genes have disparate GO terms the redundancy is close to 0. When comparing clustering results obtained by different methods, we use the value of redundancy averaged over all clusters. Furthermore, the redundancy is a very suitable measure for clustering performance comparisons because its value does not depend on the number of clusters and unlike others evaluation measures for biological network clustering [59], it is parameter-free.

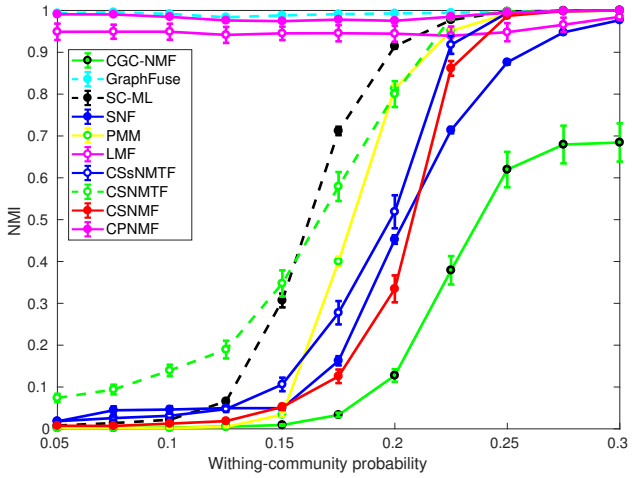

Fig. 2. The clustering performance of our proposed and other methods on 11 different *SYNTH-C* multiplex networks measured by NMI. On x-axis we present within-community probability, representing the density of connections of communities in the two complementary layers.

5 Results and Discussion

5.1 Clustering evaluation on artificial multiplex networks

The ability of our proposed methods to extract clusters from complementary layers, represented by *SYNTH-C* networks, is shown in Figure 2. The performance of our methods is compared with other methods and it is measured by NMI. By decreasing the within-community probability of complementary clusters in both layer, i.e., by decreasing the density of connections within communities and thus making communities harder to detect, we see a drastic decrease of performance in many methods, including SC-ML, PMM, SNF and CGC-NMF (Fig. 2). Furthermore, below some value of within-community probability, i.e., < 0.1, the performance of these methods is equal or close to zero. Unlike them, our proposed methods, particularly CSNMF and CPNMF show significantly better performance. Specifically, CPNMF demonstrates constant performance for all values of within-community probability. The similar results can also be observed for GraphFuse and LMF. Given that, we can observe that CPNMF method is mostly successful in utilizing complementary information contained in all layers and achieving the highest clustering results.

In terms of noise, incorporated into *SYNTH-N* networks, the ranking between the methods in terms of clustering performance is different. By increasing the between-community probability of the first layer, and thus introducing more noise between communities, the clustering performance of all methods decreases (Fig. 3). Our proposed methods, CSNMF, CSNMTF and CSsNMTF, along with SC-ML demonstrate the best performance across different values of within-community probability, which makes them more robust to noise than other methods. On the other hand, other methods methods are characterized with significantly lower clustering performance.

Out of all four methods, *CPNMF* is a method that performs reasonably good in both situations (i.e., in both

SYNTH-N and *SYNTH-C* networks). However, it is not to be expected that one single method can be equally good in in handling both noisy and complementary network layers. Furthermore, it is expected that our four methods are complementary to each other; i.e., *CSNMTF* is better in handling complementary network layers (*SYNTH-C*) than noisy network layers (*SYNTH-N*); whereas, *CSNMF* is better in handling noisy network layers (*SYNTH-N*) than complementary network layers (*SYNTH-C*). This is also reflected in the performance of our methods on the real-world networks (see Table 3), where we do not have a situation in which one single method outperforms all other methods in all data sets.

5.2 Clustering evaluation on real-world multiplex networks

Table 3 presents the Purity, NMI and ARI of our four proposed collective factorization methods, along with five different baseline methods and eight different widely used *state-of-the-art* methods on six different real-world networks. The first important observation is that all four proposed collective NMF methods (CSNMF, CPNMF, CSNMTF and CSsemi-NMTF) perform better than their corresponding baseline methods (SNMF, PNMF, SNMTF and Ssemi-NMTF) on all real-world multiplex networks. Thus, the strategy of merging layers into a single layer always leads to underperformance. Moreover, single-layer modularity maximization (MM) algorithm is outperformed by baseline, single-layer NMF methods in almost all real-world networks, except for WTN networks where MM significantly outperforms baseline NMF methods, and SND(o) where MM performs better than SNMF, SNMTF and Ssemi-NMTF, but not better than PNMF. In comparison to the *state-of-the-art* methods (PMM, SNF, SC-ML, LMF, GraphFuse, CGC-NMF, GL, Infomap), at least one of our proposed methods outperforms them all (in terms of either Purity, NMI or ARI or all three measures) in all real-world multiplex network.

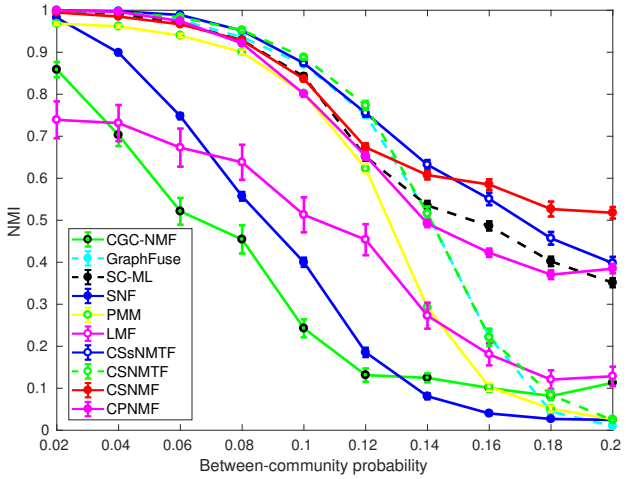

Fig. 3. The clustering performance of our proposed and other methods on 10 different *SYNTH-N* multiplex networks measured by NMI. On x-axis we present between-community probability, representing the noise level between communities in the first layer.

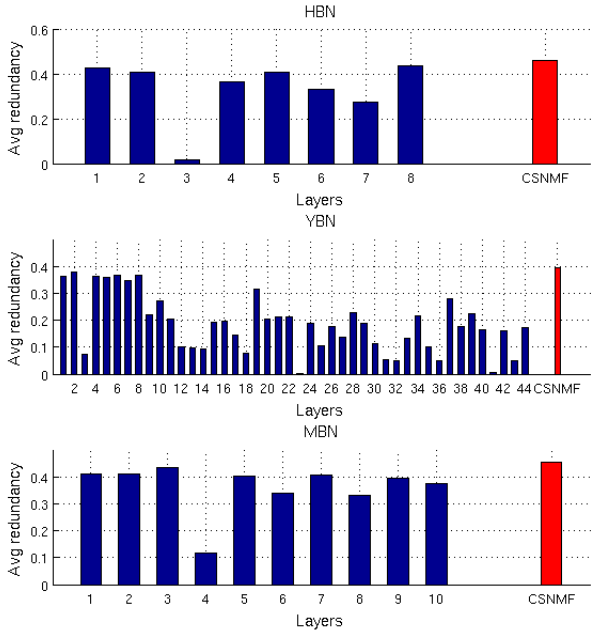

Fig. 4. Average redundancy of each individual network layer (in blue), computed by *SNMF*, and of their fused representation (in red), computed by *CSNMF*, for networks: HBN (top), YBN (middle) and MBN (bottom). The ethod parameters are: $k = 300$ and $\alpha = 0.01$.

Moreover, for example, on MPD network, both CSNMF and CSsemi-NMTF perform better than all other methods, with CSemi-NMTF being the best in terms of Purity and NMI; on SND(s) network, CSNMF, CSNMTF and CSsemi-NMTF perform better than all other methods, with CSNMTF performing the best in terms of all three measures; on WBN network, both CSNMF and CSNMTF perform better than all other methods, with CSNMTF being the best in terms of Purity and ARI, and CSNMF being the best in terms of NMI.

5.2.1 *Clustering evaluation on multiplex biological networks*

In table 4, we also present the average redundancy (R_{avg}) obtained by clustering multiplex biological networks with our four methods, as well as with *state-of-the-art* and baseline methods. The results, again, indicate the superior performance of our methods over the *state-of-the-art* and baseline methods, except in the case of MBN, where SNMTF, applied on merged network layers, yields the highest redundancy.

Furthermore, we compare the functional consistency of clusters obtain by collective integration of all network layers with the functional consistency of clusters obtained from each individual network layer. The results for all three biological networks, obtained by applying SNMF on each individual network layer and CSNMF on all network layers together, are depicted in Fig. 4. We observe that each individual network layer has biologically consistent clusters. However, the highest level of biological consistency, measured by the average redundancy, is achieved when all the layers are fused together (red bars in Fig.4).

TABLE 3
Clustering accuracy measures for methods (from left to right): MM, SNMF, PNMF, SNMTF, SsNMTF, PMM, SNF, SC-ML, LMF, GraphFuse, CGC-NMF, GL, Infomap, CSNMF, CPNMF, CSNMTF, CSsNMTF applied on real-world multi-layer networks (from top to bottom): *CiteSeer*, *CoRA*, *MPD*, *SND*, *WBN*, *WTN*, *HBN*, *YBN* and *MBN*. *AVG* row represents the average performance of the methods over all datasets.

		MM	SNMF	PNMF	SNMTF	SsNMTF	PMM	SNF	SC-ML	LMF	GF	CGC	GL	Infomap	CSNMF	CPNMF	CSNMTF	CSsNMTF
CiteSeer	Purity	0.500	0.407	0.405	0.377	0.371	0.302	0.214	0.419	0.235	0.512	0.212	**0.669**	0.509	0.501	0.519	0.416	0.404
	NMI	0.187	0.222	0.221	0.170	0.164	0.145	0.023	0.191	0.013	0.211	0.013	**0.323**	0.265	0.237	0.216	0.172	0.195
	ARI	0.152	0.059	0.057	0.042	0.038	0.008	0.001	0.169	0.005	0.201	0.001	0.155	0.157	**0.207**	0.185	0.093	0.089
CoRA	Purity	0.706	0.669	0.660	0.669	0.669	0.496	0.733	0.787	0.492	0.642	0.678	0.781	**0.922**	0.802	0.790	0.683	0.684
	NMI	0.340	0.385	0.353	0.382	0.382	0.085	0.449	0.480	0.002	0.201	0.389	0.418	0.373	**0.514**	0.480	0.346	0.390
	ARI	0.257	0.280	0.247	0.277	0.277	0.030	0.470	0.485	0.001	0.209	0.296	0.334	0.016	**0.491**	0.470	0.279	0.288
MPD	Purity	0.563	0.678	0.666	0.620	0.678	0.689	0.620	0.701	0.471	0.689	0.678	0.678	0.701	0.701	0.655	0.655	**0.724**
	NMI	0.313	0.471	0.466	0.384	0.473	0.533	0.395	0.495	0.191	**0.565**	0.457	0.467	0.410	0.504	0.451	0.458	0.521
	ARI	0.147	0.268	0.259	0.228	0.272	0.383	0.280	0.379	0.029	0.411	0.357	0.372	0.115	0.394	0.368	0.346	**0.422**
SND(o)	Purity	0.929	**0.943**	**0.943**	0.676	**0.943**	**0.943**	**0.943**	**0.943**	0.788	**0.943**	**0.943**	0.929	0.676	**0.943**	**0.943**	**0.943**	**0.943**
	NMI	0.618	0.681	0.681	0.133	0.681	0.675	0.689	0.681	0.303	0.675	0.673	0.618	0.001	0.681	0.685	**0.773**	0.678
	ARI	0.460	0.493	0.493	0.021	0.493	0.477	0.515	0.493	0.239	0.477	0.472	0.460	0.012	0.493	0.503	**0.811**	0.484
SND (s)	Purity	0.619	0.577	0.633	0.634	0.619	0.591	0.633	0.591	0.633	0.619	0.662	0.671	0.507	0.633	0.633	0.747	0.605
	NMI	0.038	0.025	0.052	0.055	0.041	0.037	0.057	0.030	0.053	0.045	0.0781	0.097	0.001	0.053	0.053	**0.276**	0.034
	ARI	0.024	0.012	0.058	0.058	0.043	0.022	0.058	0.021	0.059	0.044	0.092	0.089	0.001	0.059	0.059	**0.234**	0.031
WBN	Purity	0.473	0.512	0.501	0.476	0.523	0.473	0.534	0.272	0.283	0.509	0.516	0.566	0.401	**0.577**	0.537	0.548	0.530
	NMI	0.333	0.382	0.400	0.327	0.363	0.373	0.425	0.079	0.098	0.426	0.370	0.398	0.355	**0.463**	0.432	0.404	0.424
	ARI	0.199	0.226	0.213	0.112	0.180	0.130	0.211	0.001	0.009	0.216	0.211	0.185	0.200	**0.291**	0.233	0.237	0.225
WTN	Purity	0.506	0.475	0.464	0.388	0.284	0.388	0.289	0.497	0.453	0.415	0.278	**0.783**	0.453	0.579	0.420	0.371	0.420
	NMI	0.231	0.269	0.242	0.176	0.077	0.205	0.073	0.226	0.191	0.176	0.072	**0.426**	0.273	0.322	0.172	0.154	0.155
	ARI	0.080	0.114	0.114	0.073	0.001	0.039	0.005	0.133	0.094	0.107	0.002	0.068	0.273	**0.160**	0.094	0.035	0.088
AVG	Purity	0.613	0.608	0.610	0.548	0.583	0.554	0.566	0.601	0.479	0.618	0.566	**0.725**	0.595	0.676	0.642	0.6233	0.6157
	NMI	0.294	0.347	0.345	0.232	0.311	0.293	0.301	0.311	0.121	0.328	0.293	0.392	0.239	**0.396**	0.355	0.369	0.342
	ARI	0.188	0.207	0.205	0.115	0.186	0.155	0.220	0.240	0.062	0.237	0.204	0.237	0.110	**0.300**	0.273	0.290	0.232

5.3 Contribution of different network types to clustering performance

We have seen in Fig. 4 that different network layers carry different information and thus the clustering of their nodes results in different performance. In order to estimate the contribution of different network layers to the final, consensus clustering performance and to distinguish between more and less informative network layers, we compute the distance between the consensus low-dimensional feature representation, $\mathbf{H}$, and the low-dimensional feature representation of individual network layers, $\mathbf{H}^{(i)}$, as introduced in equation 5. We hypotheses that the network layers with the low-dimensional feature representations that are the larger distance from the consensus feature representation contribute less than those that are at the smaller distance from the consensus feature representation. The results for biological multiplex networks are shown in Fig. 5. Comparing these results with the ones shown in Fig. 4, we can see that, for example, in HBN, layer 3 is the least informative (Fig. 4 (top)) and also with the largest projection distance (Fig. 5 (top)). Furthermore, in YBN we observe that the two least informative layers are layer 23 and layer 41 (Fig. 4 (middle)) are also those with the largest distance (Fig. 5 (middle). Similar observation holds for MBN. This is an interesting observation and it suggests that the projection distance could be used as an indicator for selecting more or less informative network layers even before the clustering is performed.

TABLE 4
Clustering accuracy measures for methods (from top to bottom): MM, SNMF, PNMF, SNMTF, SsNMTF, PMM, SNF, SC-ML, LMF, GraphFuse, CGC-NMF, GL, Infomap, CSNMF, CPNMF, CSNMTF, CSsNMTF applied on biological multi-layer networks (from left to right): *HBN*, *YBN* and *MBN*

Method	HBN	YBN	MBN
MM	0.180	0.027	0.015
SNMF	0.325	0.374	0.416
PNMF	0.350	0.336	0.387
SNMTF	0.322	0.372	**0.462**
SsNMTF	0.326	0.358	0.441
PMM	0.351	0.343	0.355
SNF	0.141	0.163	0.211
SC-ML	0.266	0.257	0.320
LMF	0.045	0.100	0.180
GraphFuse	0.203	0.005	0.298
CGC-NMF	0.320	0.085	0.002
GL	0.161	0.114	0.014
Infomap	0.172	0.311	0.328
CSNMF	0.341	**0.383**	0.422
CPNMF	**0.364**	0.342	0.401
CSNMTF	0.339	**0.383**	0.416
CSsNMTF	0.342	0.381	0.433

6 Conclusion

In this paper, we address the problem of composite community detection in multiplex networks by proposing NF-CCE, a general model consisting of four novel methods, *CSNMF*, *CPNMF*, *CSNMTF* and *CSsemi-NMTF*, based on four non-negative matrix factorization techniques. Each of the proposed method works in a similar way: in the first step, it decomposes adjacency matrices representing network layers into low-dimensional, non-negative feature matrices; then, in the second step, it fuses the feature matrices of layers into a consensus non-negative, low-dimensional feature matrix common to all network layers, from which the composite clusters are extracted. The second step is done by collective matrix factorization that maximizes the shared information between network layers by optimizing the distance between each of the non-negative feature matrices representing layers and the consensus feature matrix.

The ability of our methods to integrate complementary as well as noisy network layers more efficiently than the *state-of-the-art* methods has been demonstrated on artificially generated multiplex networks. In terms of cluster-

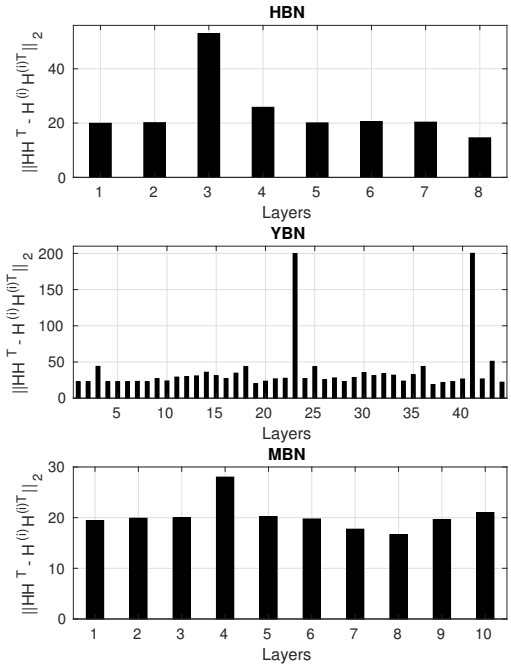

Fig. 5. Projection distance between the networks' low-dimensional consensus representation and the representation of its individual layers. The distance is computed by Eq. 5, and the low-dimensional representation by *CSNMF* method with parameters $k = 300$ and $\alpha = 0.01$. The results are shown for HBN (top), YBN (middle) and MBN (bottom) multiplex biological networks.

ing accuracy, we demonstrate the superior performance of our proposed methods over the baseline and *state-of-the-art* methods on nine real-world networks. We show that simple averaging of adjacency matrices representing network layers (i.e., merging network layers into a single network representation), the strategy that is usually practiced in the literature, leads to the worst clustering performance. Moreover, our experiments indicate that widely-used modularity maximization methods are significantly outperformed by NMF-based methods.

NF-CCE can be applied on multiplex networks from different domains, ranging from social, phone communication and bibliographic networks to biological, economic and brain networks, demonstrating the diverse applicability of our methods.

REFERENCES

[1] M. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc., 2010.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 45, pp. 175 – 308, 2006.

[3] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, pp. 268–276, 2001.

[4] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.

[5] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[6] J. B. Pereira-Leal, A. J. Enright, and C. A. Ouzounis, "Detection of functional modules from protein interaction networks," *Proteins: Structure, Function, and Bioinformatics*, vol. 54, no. 1, pp. 49–57, 2004.

[7] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 631–640.

[8] J. Chen and B. Yuan, "Detecting functional modules in the yeast proteinprotein interaction network," *Bioinformatics*, vol. 22, no. 18, pp. 2283–2290, 2006.

[9] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Phys. Rev. E*, vol. 72, p. 027104, Aug 2005.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[11] M. Mitrović and B. Tadić, "Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities," *Phys. Rev. E*, vol. 80, p. 026123, Aug 2009.

[12] M. A. Porter, J.-P. Onnela, and P. J. Mucha, "Communities in networks," *Notices Amer. Math. Soc.*, vol. 56, no. 9, pp. 1082–1097, 2009.

[13] S. E. Schaeffer, "Survey: Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, Aug. 2007.

[14] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, and A. Arenas, "Mathematical formulation of multilayer networks," *Phys. Rev. X*, vol. 3, p. 041022, Dec 2013.

[15] V. Nicosia, G. Bianconi, V. Latora, and M. Barthelemy, "Growing multiplex networks," *Phys. Rev. Lett.*, vol. 111, p. 058701, Jul 2013.

[16] 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, 2010.

[17] J. Kim and J.-G. Lee, "Community detection in multi-layer graphs: A survey," *SIGMOD Rec.*, vol. 44, no. 3, pp. 37–48, dec 2015.

[18] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of Complex Networks*, 2014.

[19] D. Kuang, H. Park, and C. H. Ding, "Symmetric nonnegative matrix factorization for graph clustering." in *SDM*, vol. 12. SIAM, 2012, pp. 106–117.

[20] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, "Community discovery using nonnegative matrix factorization," *Data Min. Knowl. Discov.*, vol. 22, no. 3, pp. 493–521, May 2011.

[21] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering on multi-layer graphs via subspace analysis on grassmann manifolds," *Trans. Sig. Proc.*, vol. 62, no. 4, pp. 905–918, feb 2014.

[22] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, feb 1998.

[23] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 576–588, March 2010.

[24] W. Cheng, X. Zhang, Z. Guo, Y. Wu, P. F. Sullivan, and W. Wang, "Flexible and robust co-regularized multi-domain graph clustering," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 320–328.

[25] L. Tang, X. Wang, and H. Liu, *Uncovering groups via heterogeneous interaction analysis*, 2009, pp. 503–512.

[26] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with multi-layer graphs: A spectral perspective," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5820–5831, Nov 2012.

[27] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *In NIPS*. MIT Press, 2000, pp. 556–562.

[28] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.

[29] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[30] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[31] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[32] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

[33] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109 – 137, 1983.

[34] M. G. Everett and S. P. Borgatti, "Analyzing clique overlap," *Connections*, vol. 21, no. 1, pp. 49–61, 1998.

[35] S. Zhang, H. Zhao, and M. K. Ng, "Functional module analysis for gene coexpression networks with network integration," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 12, no. 5, pp. 1146–1160, Sep. 2015.

[36] M. Berlingerio, M. Coscia, and F. Giannotti, "Finding and characterizing communities in multidimensional networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, July 2011, pp. 490–494.

[37] M. A. Rodriguez and J. Shinavier, "Exposing multi-relational networks to single-relational network analysis algorithms," *Journal of Informetrics*, vol. 4, no. 1, pp. 29 – 41, 2010.

[38] I. S. Jutla, L. G. Jeub, and P. J. Mucha, "A generalized louvain method for community detection implemented in matlab," *URL http://netwiki. amath. unc. edu/GenLouvain*, 2011.

[39] B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe-Kains, and A. Goldenberg, "Similarity network fusion for aggregating data types on a genomic scale," *Nature Methods*, vol. 11, no. 3, pp. 333–337, 2014.

[40] 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, p. 011027, Mar 2015.

[41] Y. Wang and X. Qian, "Biological network clustering by robust nmf," in *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, ser. BCB '14. New York, NY, USA: ACM, 2014, pp. 688–689.

[42] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ser. ICDM '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1016–1021.

[43] E. E. Papalexakis, L. Akoglu, and D. Ienco, "Do more views of a graph help? community detection and clustering in multi-graphs," in *Proceedings of the 16th International Conference on Information Fusion, FUSION 2013, Istanbul, Turkey, July 9-12, 2013*, 2013, pp. 899–905.

[44] L. Gauvin, A. Panisson, and C. Cattuto, "Detecting the community structure and activity patterns of temporal networks: A non-negative tensor factorization approach," *PLOS ONE*, vol. 9, no. 1, pp. 1–13, 01 2014.

[45] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, no. 1, p. 84, 1970.

[46] Z. Yang and E. Oja, "Linear and nonlinear projective nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 21, no. 5, pp. 734–749, May 2010.

[47] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix T-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 126–135.

[48] C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, Jan 2010.

[49] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[50] A. Condon and R. M. Karp, "Algorithms for graph partitioning on the planted partition model," *Random Struct. Algorithms*, vol. 18, no. 2, pp. 116–140, Mar. 2001.

[51] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[52] T. A. B. Snijders, P. E. Pattison, G. L. Robins, and M. S. Handcock, "New specifications for exponential random graph models," *Sociological Methodology*, vol. 36, no. 1, pp. 99–153, 2006.

[53] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, "The structure of the nervous system of the nematode caenorhabditis elegans," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 314, no. 1165, pp. 1–340, 1986.

[54] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature communications*, vol. 6, p. 6864, 2015.

[55] S. Mostafavi and Q. Morris, "Fast integration of heterogeneous data sources for predicting gene function with limited annotation," *Bioinformatics*, vol. 26, no. 14, pp. 1759–1765, 2010.

[56] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Mach. Learn.*, vol. 55, no. 3, pp. 311–331, jun 2004.

[57] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[58] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, May 2000.

[59] Y.-K. Shih and S. Parthasarathy, "Identifying functional modules in interaction networks through overlapping markov clustering," *Bioinformatics*, vol. 28, no. 18, pp. i473–i479, 2012.

Vladimir Gligorijević is a Research Fellow in the Center for Computational Biology (CCB) at the Flatiron Institute, Simons Foundation, USA. He received his BSc and MSc degrees from the Department of Physics, University of Belgrade, Serbia, and PhD from the Department of Computing, Imperial College London, UK. His research interests span the areas of machine learning and network theory with applications in social sciences, computational biology and bioinformatics.

Yannis Panagakis is a Lecturer (Assistant Professor equivalent) in Computer Science at Middlesex University London and Research Faculty at Imperial College London (Department of Computing). His research interests lie in machine learning and its interface with signal processing, high-dimensional statistics, and computational optimization. Specifically, Yannis is working on statistical models and algorithms for robust and efficient learning from high-dimensional data and signals conveying audio, visual, behavioural, medical, and social information. He received his PhD and MSc degrees from the Department of Informatics, Aristotle University of Thessaloniki and his BSc degree in Informatics and Telecommunication from the University of Athens, Greece. Yannis has been awarded the prestigious Marie-Curie Fellowship, among various scholarships and awards for his studies and research. His work has been published leading journals and conferences proceedings, including IEEE T-PAMI, IJCV, CVPR, and ICCV. He currently serves as the Managing Editor-in-Chief of Image and Vision Computing Journal. Dr Panagakis co-organised the British Machine Vision Conference (BMVC) in 2017 and numerous workshops in conjunction with international conferences.

Stefanos Zafeiriou is currently a Reader in Pattern Recognition/Statistical Machine Learning for Computer Vision with the Department of Computing, Imperial College London, London, U.K, and a Distinguishing Research Fellow with University of Oulu under Finish Distinguishing Professor Programme. He was a recipient of the Prestigious Junior Research Fellowships from Imperial College London in 2011 to start his own independent research group. He was the recipient of the Presidents Medal for Excellence in Research Supervision for 2016. He has received various awards during his doctoral and post-doctoral studies. He currently serves as an Associate Editor of the IEEE Transactions on Cybernetics the Image and Vision Computing Journal. He has been a Guest Editor of over six journal special issues and co-organised over nine workshops/special sessions on face analysis topics in top venues, such as CVPR/FG/ICCV/ECCV (including two very successfully challenges run in ICCV13 and ICCV15 on facial landmark localisation/tracking). He has more than 5000 citations to his work, h-index 37. He is the General Chair of BMVC 2017.