ORTHOGONAL NONNEGATIVE MATRIX TRI-FACTORIZATION FOR COMMUNITY DETECTION IN MULTIPLEX NETWORKS

Meiby Ortiz-Bouza and Selin Aviyente

Department of Electrical and Computer Engineering, Michigan State University
East Lansing, 48824
aviyente@egr.msu.edu

ABSTRACT

Networks provide a powerful tool to model complex systems. Recently, there has been a growing interest in multiplex networks as they can represent the interactions between a pair of nodes through multiple types of links, each reflecting a distinct type of interaction. One of the important tools in understanding network topology is community detection. Existing work on multiplex community detection mostly focuses on learning a common community structure across layers without taking the heterogeneity of the different layers into account. In this paper, we introduce a new multiplex community detection approach that can identify communities that are common across layers as well as those that are unique to each layer. The proposed algorithm employs Orthogonal Non-Negative Matrix Tri-Factorization to model each layer's adjacency matrix as the sum of two low-rank matrix factorizations, corresponding to the common and private communities, respectively. The proposed algorithm is evaluated on both synthetic and real multiplex networks and compared to state-of-the-art techniques.

***Index Terms*—** Multiplex Networks, Community Detection, Nonnegative Matrix Tri-factorization, Low Rank Structure

1. INTRODUCTION

Many real world systems, including social and biological ones, are often represented as complex networks capturing the interactions between multiple agents [1]. The different agents are represented as the nodes of the network, and the relations among them are encoded by the edges of the network. However, traditional network models that employ simple graphs cannot capture the diverse nature of the connectivity patterns between entities, i.e., multiple types of interactions. For this reason, 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 but may have very different topology [2]. This model has been used to study a large variety of systems across disciplines, ranging from living organisms and human societies to transportation systems and critical infrastructures [3, 4].

An important aspect of network analysis is the discovery of communities defined as groups of nodes that are more densely connected to each other than they are to the rest of the network. While a large body of work exists on community detection [5], most of it is focused on single layer networks. Existing community detection algorithms for multiplex networks can be grouped into three main classes. The first class of methods consists of simplifying the multiplex network into a graph by merging its layers, using a so-called flattening algorithm, then applying a traditional community detection algorithm [6, 7, 8]. While these methods are computationally efficient, the algorithms in this class are only able to identify communities that are common across all layers, and some spurious communities may emerge because of the flattening process. The second class of methods is layer-by-layer methods, where traditional community detection methods are applied to each layer individually and the results are then merged. Some examples of algorithms under this class are ABACUS [9], PMM [10], and SC-ML [11]. As a consequence of the layer-by-layer community detection step, these methods include nodes in the same community only when they are part of the same community in at least one layer. Finally, the third class of algorithms operates directly on the multiplex network model. Some examples include random walk based algorithms such as LART [12] and Infomap [13] and multilayer community quality metric optimization based methods [14, 15, 16] such as Generalized Louvain [15]. The majority of existing multiplex community detection approaches identify a partition that best fits all given layers. However, none of these methods detect communities that are common across layers as well as those unique to each layer, simultaneously. This is particularly important in real world applications where the different layers correspond to different modes of interaction and the networks are heterogeneous. For example in social networks, a group of individuals may be well connected via friendships on Facebook; however, this common group of actors will likely, for example, not work at the same company. In realistic situations such as these, a given vertex 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, we introduce a novel framework titled Multiplex Orthogonal Non-negative Matrix Trifactorization (MX-ONMTF) for detecting communities that are common across layers as well as communities that are unique to each layer, i.e., private communities. The proposed approach relies on the principle of minimizing the normalized cut using Non-negative Matrix Factorization (NMF) [17]. Each layer of the multiplex network is modeled as the sum of two low-rank matrix factorizations where the first term corresponds to the common communities and the second term corresponds to the private communities. The resulting joint optimization problem is solved using an iterative multiplicative update algorithm.

2. BACKGROUND

2.1. Multiplex Networks

Multiplex networks can be represented using a finite sequence of graphs $\{G_l\}$, $l \in \mathbf{L}$, where $G_l = (V_l, E_l)$, $\mathbf{L} = \{1, 2, ..., L\}$ is the set of layers, and $V_l \subseteq \{1, 2, ..., n\}$ is the set of nodes in layer l

This work was supported in part by National Science Foundation under Grants CCF-2006800.

[18]. Each graph $G_l = (V_l, E_l)$ can be represented by an adjacency matrix, $A_l \in R^{n\times n}$, $l \in \mathbf{L}$.

2.2. Orthogonal Non-Negative Matrix Tri-Factorization

NMF [19] decomposes a nonnegative matrix $V \in R^{n\times m}$ into the product of two low-rank nonnegative matrices $W \in R^{n\times k}$ and $U \in R^{m\times k}$, such that $V \approx WU^T$ and $k \ll n, m$. W and U are found by solving the optimization problem

$$\underset{W\geq,U\geq 0}{\operatorname{argmin}} ||V - WU^T||_F^2.$$

For NMF based community detection algorithms, W and U are the community feature matrix and the community indicator matrix, respectively. Adding orthogonality constraints to either or both factor matrices, $W^TW = I$ and/or $U^TU = I$, improves the performance of NMF as orthogonality and nonnegativity force each row of W (U) to have only one nonzero element which implies that each node belongs only to one community (hard clustering). Orthogonal NMF has been used in community detection where V is usually the adjacency matrix and k is the number of communities [20, 21].

There are several extensions of the original NMF including Symmetric Non-Negative Matrix Tri-factorization [22] (SNMTF), where V is approximated by USU^T. The symmetric matrix $S \in R^{k\times k}$ provides more degrees of freedom increasing the accuracy of the approximation while U provides community membership information. In this paper, we will use Orthogonal Non-negative Matrix Tri-factorization (ONMTF), which is an extension of SNMTF with orthogonality constrains on U, to formulate the community detection problem in multiplex networks.

3. PROPOSED METHOD (MX-ONMTF)

The proposed method, MX-ONMTF, models each layer's adjacency matrix as a sum of low-dimensional representations of common and private communities using ONMTF.

3.1. Problem Formulation

For a two-layer multiplex network with adjacency matrices, $A_1 \in R^{n\times n}$ and $A_2 \in R^{n\times n}$, the proposed model based on ONMTF can be formulated as

$$\begin{aligned}&\underset{H\geq 0, H_l\geq 0, S_l\geq 0, G_l\geq 0}{\operatorname{argmin}} \sum_{l=1}^{2} ||A_l - HS_lH^T - H_lG_lH_l^T||_F^2 \\ &\text{s.t. } H^TH = I, H_1^TH_1 = I, H_2^TH_2 = I, l \in \{1,2\},\end{aligned} \tag{1}$$

where $H \in R^{n\times k_c}$ and $H_l \in R^{n\times k_{p_l}}$ are the community membership matrices corresponding to the common and private communities, respectively, and S_1, S_2, G_1 and G_2 are symmetric matrices. In this formulation, it is assumed that there are k_c common communities across the two layers, $kp_1 = k_1 - k_c$ private communities in layer 1 and $kp_2 = k_2 - k_c$ private communities in layer 2, where k_1 and k_2 are the number of communities in layers 1 and 2, respectively.

3.2. Optimization solution

The optimization problem in (1) is not convex for all variables H, H_l, S_l, and G_l, simultaneously. It can be solved using a multiplicative update algorithm (MUA) [22] where, during each iteration, a variable is updated while the rest are fixed. Multiplicative update algorithms for solving NMTF with orthogonal constraints was first addressed by [22]. In this paper, we follow a similar approach to find the multiplicative update rules for each variable.

To find the update rules for H, H_l, S_l, and G_l, we optimize the constrained problem introducing Lagrange multipliers Λ and Λ_l and minimizing the following Lagrangian function:

$$\begin{aligned}\mathcal{L}(H, H_l, S_l, G_l) &= \sum_{l=1}^{2} ||A_l - HS_lH^T - H_lG_lH_l^T||_F^2 \\ &+ tr(\Lambda(H^TH - I)) + \sum_{l=1}^{2} tr(\Lambda_l(H_l^TH_l - I)).\end{aligned} \tag{2}$$

For updating H, we find $\nabla_H \mathcal{L}$ as

$$\begin{aligned}\nabla_H \mathcal{L} &= 4HS_1^TH^THS_1 + 4H_1G_1^TH_1^THS_1 - 4A_1HS_1 \\ &+4HS_2^TH^THS_2 + 4H_2G_2^TH_2^THS_2 - 4A_2HS_2 + 4H\Lambda.\end{aligned} \tag{3}$$

Applying the KKT conditions $\nabla_H \mathcal{L} = 0$ and $\nabla_\Lambda \mathcal{L} = 0$, we obtain:
(i) $\Lambda = \sum_{l=1}^{2}(-S_l^TS_l - H^TH_lG_l^TH_l^THS_l + H^TA_lHS_l)$.
(ii) $H^TH = I$.
Substituting (i) and (ii) in (3), we get

$$\begin{aligned}\nabla_H \mathcal{L} = &\sum_{l=1}^{2}(4H_lG_l^TH_l^THS_l - 4A_lHS_l + 4HH^TA_lHS_l \\ &-4HH^TH_lG_l^TH_l^THS_l).\end{aligned} \tag{4}$$

As discussed in [23], if the gradient of an error function, ε, is of the form $\nabla\varepsilon = \nabla\varepsilon^+ - \nabla\varepsilon^-$, where $\nabla\varepsilon^+ > 0$ and $\nabla\varepsilon^- > 0$, then the multiplicative update for parameter Θ has the form $\Theta = \Theta * \frac{\nabla\varepsilon^-}{\nabla\varepsilon^+}$. It can be easily seen that the multiplicative update preserves the non-negativity of Θ, while $\nabla\varepsilon = 0$ when the convergence is achieved. Following this procedure, from the gradient obtained in Eq. (4) of the error function, we derive the following multiplicative update rule for H

$$H \leftarrow H * \frac{\sum_{l=1}^{2}(A_lHS_l + HH^TH_lG_l^TH_l^THS_l)}{\sum_{l=1}^{2}(H_lG_l^TH_l^THS_l + HH^TA_lHS_l)}, \tag{5}$$

where the multiplication and division are performed element-wise and both the numerator and denominator are positive. The update rules for H_1, H_2, S_1, S_2, G_1 and G_2 can be obtained in a similar manner as follows:

$$H_l \leftarrow H_l * \frac{A_lH_lG_l + H_lH_l^THS_l^TH^TH_lG_l}{HS_l^TH^TH_lG_l^T + H_lH_l^TA_lH_lG_l}, \tag{6}$$

$$S_l \leftarrow S_l * \frac{H^TA_lH}{H^THS_lH^TH + H^TH_lG_lH_l^TH}, \tag{7}$$

$$G_l \leftarrow G_l * \frac{H_l^TA_lH_l}{H_l^TH_lG_lH_l^TH_l + H_l^THS_lH^TH_l}, \tag{8}$$

for $l \in \{1,2\}$. This formulation can be easily extended to L layers.

3.3. Finding the number of communities

In most NMF-based community detection algorithms, the number of communities (k) is an input parameter. Usually this problem is addressed by detecting communities with different values of k and then selecting the one that gives the best results in terms of a predetermined quality function such as modularity.

In this paper, we propose a two-step approach to determine the number of communities per layer as well as the number of common communities. First, we find the number of communities in each layer (k_1 and k_2) using the eigengap rule. Next, we apply ONMTF to each layer to obtain the individual community structure. We then compute the correlation of the low-rank embedding matrices, i.e., the community membership matrices, across layers to detect the number of common communities, k_c. k_c is determined as the number of positive elements of the $k_1 \times k_2$ correlation matrix that are higher than a pre-determined threshold.

3.4. Algorithm Implementation

The pseudocode for implementing the proposed optimization approach is given in Algorithm 1. The algorithm first randomly initializes H, H_1, H_2, S_1, S_2, G_1 and G_2 and then proceeds to update them using Eqs. (5)-(8) over 1000 iterations or until convergence.

Algorithm 1 MX-ONMTF

Input: Adjacency matrices A_1, A_2
Number of common and per layer communities k_c, k_1, k_2
Output: Community indicator matrices H, H_1, and H_2. Community membership vector $idx \in R^n$.

for r=1 to 50 **do**
  Randomly initialize $H \geq 0, H_l \geq 0, S_l \geq 0, G_l \geq 0$ and S_l and G_l to be symmetric.
  for 1000 iterations or until convergence **do**
    update H according to Eq. (5)
    update H_l for each $l \in \{1, 2\}$ according to Eq. (6)
    update S_l for each $l \in \{1, 2\}$ according to Eq. (7)
    update G_l for each $l \in \{1, 2\}$ according to Eq. (8)
  end for
  for each node i in each layer l **do**
    $j^* = argmax_j H_{ij}$
    if $H_{ij^*} > \max\{H_{1ij^*}, H_{2ij^*}\}$ **then**
      $idx(i) = j^*$
    else
      $idx(i) = argmax_j H_{lij}$
    end if
  end for
  Compute NMI_r or Q_{Dr}.
end for
Choose the partition r^* such that $r^* = argmax_r NMI$ ($r^* = argmax_r Q_D$).

Since the algorithm is initialized using random matrices, we repeat the algorithm 50 times and select the solution that yields the maximum value of the performance metric, Normalized Mutual Information (NMI) [24] for synthetic networks with available ground truth; Modularity Density (Q_D) [25] for real networks or other networks for which ground truth information is not available.

3.5. Time complexity

The time complexity of the proposed algorithm is mostly due to the Multiplicative Update Rules, Eqs. (5)-(8). The time complexity of multiplying a $m \times k$ matrix with a $k \times n$ matrix, is $O(mkn)$. Therefore, the time complexities of (5)-(8) are $O(n^2(k_c + k_{p_1} + k_{p_2}))$, $O(n^2(k_c + k_{p_l}))$, $O(n^2 k_c)$, and $O(n^2 k_{p_l})$, respectively. Therefore, the total complexity is $O(n^2 k)$, where $k = k_c + k_{p_1} + k_{p_2}$.

4. EXPERIMENTS

In this section, we present the experimental results of our method evaluated on both a synthetic multiplex network, Multiplex Benchmark Network described in [26], and a real-world multiplex network, Lazega Law Firm Social Network [27].

The results of the generated synthetic networks are evaluated using NMI. For real-world networks, we use the Modularity Density, Q_D, defined for a given network with vertex set V as [25]:

$$Q_D(\{V_{c=1}^k\}) = \sum_{c=1}^{k} \frac{L(V_c, V_c) - L(V_c, \overline{V_c})}{|V_c|}, \tag{9}$$

where $\{V_{c=1}^k\}$ is a hard partition of the network and V_c is the set of vertices in the cth community, $L(V_c, V_c) = \sum_{i,j \in V_c} A_{ij}$ and $L(V_c, \overline{V}_c) = \sum_{i \in V_c, j \in \overline{V}_c} A_{ij}$ where $\overline{V}_c = V - V_c$.

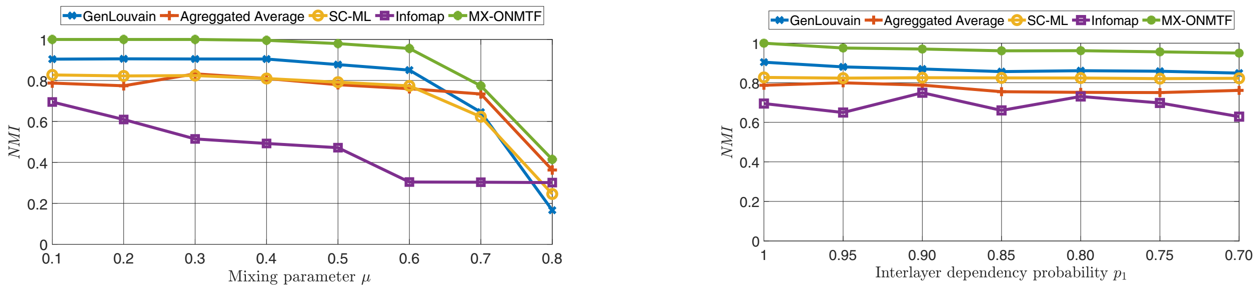
4.1. Benchmark Multiplex Networks

We generated a multiplex network with 256 nodes and two layers based on the model described in [26]. Two common communities across the two layers are generated by selecting 100 nodes at random and setting the inter-layer dependency probability to p_1. These 100 nodes are randomly assigned to two common communities. The remaining nodes are assigned randomly to private communities with inter-layer dependency probability $p_2 < 1$ so that the private communities in each layer are different. Four private communities in layer 1 and three private communities in layer 2 were generated, for a total of six and five communities in layers 1 and 2, respectively. Using this configuration, we generated two sets of simulations. In the first simulation, we evaluated the performance of our algorithm for different noise levels by generating networks with varying values of the mixing parameter $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, which controls the density of inter-community edges, and fixed inter-layer dependency probabilities, $p_1 = 1$ and $p_2 = 0.2$ to generate the common and the private communities, respectively. In the second simulation, we evaluated the robustness of the algorithm against variations in the common community structure by generating networks with $\mu = 0.1$ and varying inter-layer dependency probability, p_1, i.e., the common communities are allowed to vary across layers.

We compared the performance of our method to well-known multiplex community detection algorithms from the three different groups mentioned in Section 1, i.e., ONMTF applied to the aggregated multiplex networks (Aggregated Average), Spectral Clustering on Multi-Layer graphs (SC-ML) [11], Generalized Louvain (GenLouvain) [28], and Infomap [13].

Fig. 1a shows the average NMI over 100 realizations of the network with $p_1 = 1$ and varying μ. As seen in Fig. 1a, our method outperforms the other four methods. As GenLouvain assigns each node-layer tuple to its own community, it cannot identify common communities across layers. On the other hand, ONMTF applied to the aggregated network, SC-ML, and Infomap force each physical node across layers to the same community, thus cannot differentiate the differences across layers. Moreover, our method is more robust to noise as indicated by high NMI values for increasing μ.

The performance of all methods for networks with $\mu = 0.1$ and varying p_1 are reported in Fig. 1b based on the mean of the NMI over 100 realizations of the network. As we can see in Fig. 1b, our method still outperforms the other four methods when there is some variation in the common community across layers. This demonstrates that our method is robust to variations of the common community structure across layers.



(a) (b)

Fig. 1: Average NMI of the different methods over 100 realizations of benchmark networks: (1a) Varying values of the mixing parameter μ and common communities generated with interlayer dependency probability $p_1 = 1$ and (1b) Varying values of p_1 and fixed mixing parameter $\mu = 0.1$.

Method	Status	Gender	Office	Seniority	Age	Practice	Law school
Aggregated Average	0.0489	0.0318	0.5171	0.0956	0.0446	**0.5723**	0.0048
SC-ML	0.0280	0.0318	0.5828	0.0920	0.0543	**0.5723**	0.0051
GenLouvain	0.0350	0.0312	0.5058	0.0727	0.0395	0.5515	0.0060
Infomap	0.0192	0.0058	0.1815	0.2889	0.0071	0.0005	0.0103
MX-ONMTF	**0.3902**	**0.4100**	**0.7297**	**0.2191**	**0.3024**	0.5610	**0.3308**

Table 1: NMI scores between node attributes and the community structured detected by MX-ONMTF for the Lazega Law Firm data set.

Method	Run Time (seconds)
Aggregated Average	0.00015
SC-ML	0.2636
GenLouvain	0.0184
Infomap	0.7359
MX-ONMTF	0.5343

Table 2: Average run times for different community detection methods. The average time was computed over 100 runs.

Table 2 reports the average run times for the different algorithms considered in this paper. NMF applied to aggregated average is the fastest method as it is applied to a single layer network and is not a multiplex community detection method. Among the multiplex community detection methods, GenLouvain is the fastest method as its complexity is known to be $\mathcal{O}(n \log n)$. MX-ONMTF is faster than InfoMap and comparable to SC-ML.

4.2. Lazega Law Firm Multiplex Network

Lazega Law Firm [27] 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. For this paper, we only used the layers representing Co-work (layer 1) and Advice (layer 2) relationships. This data set also includes metadata of some attributes of each node such as status (partner or associate), gender, office location (Boston, Hartford, or Providence), years with the firm, age, practice (litigation or corporate), and law school (Harvard, Yale, UConn or other).

Applying MX-ONMTF on this network, we obtained one common community across layers 1 and 2 composed of 19 nodes colored in blue as well as private communities for each layer as shown in Fig. 2. Since this network does not have ground truth community structure, we compute the NMI between the detected community structure and each type of node attributes to gain better insight

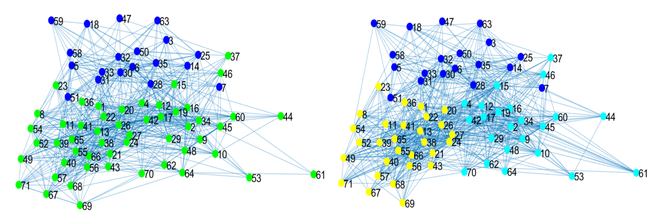


Fig. 2: Common community (blue) across layers 1 and 2 and the private communities detected by MX-ONMTF.

to the results. As we can see in Table 1, our method has the highest NMI values for each of the attributes, except for Practice where Aggregated Average and SC-ML have slightly better NMI. However, Aggregated Average and SC-ML perform considerably worse for the rest of the attributes. This shows that our method achieves a trade-off by detecting community structures that capture all attributes instead of partitioning the network with respect to just one attribute.

5. CONCLUSIONS

In this paper, we proposed a community detection method for multiplex networks based on ONMTF. The proposed method, MX-ONMTF, detects both common and private communities across layers, allowing us to discover the full community structure. We also proposed a new approach for determining the number of communities reducing the computational complexity of conventional methods which perform a greedy search over a range of k values. Results for both synthetic and real-world networks show that our method is superior to existing methods as it does not enforce a consensus community structure across layers while differentiating between common and private communities. The latter is important for real networks where there is heterogeneity in the relationships across layers. Future work will consider extensions to more than two layers.

6. REFERENCES

[1] A.-L. Barabási, "Network science," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1987, p. 20120375, 2013.

[2] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," vol. 2, no. 3, pp. 203–271, 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, 2019.

[4] A. Aleta, S. Meloni, and Y. Moreno, "A multilayer perspective for the analysis of urban transportation systems," *Scientific reports*, vol. 7, no. 1, pp. 1–9, 2017.

[5] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics reports*, vol. 659, pp. 1–44, 2016.

[6] M. Berlingerio, M. Coscia, and F. Giannotti, "Finding and characterizing communities in multidimensional networks," in *2011 International Conference on advances in social networks analysis and mining*. IEEE, 2011, pp. 490–494.

[7] P.-Y. Chen and A. O. Hero, "Multilayer spectral graph clustering via convex layer aggregation: Theory and algorithms," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 553–567, 2017.

[8] D. Taylor, R. S. Caceres, and P. J. Mucha, "Super-resolution community detection for layer-aggregated multilayer networks," *Physical Review X*, vol. 7, no. 3, p. 031056, 2017.

[9] M. Berlingerio, F. Pinelli, and F. Calabrese, "Abacus: frequent pattern mining-based community discovery in multidimensional networks," *Data Mining and Knowledge Discovery*, vol. 27, no. 3, pp. 294–320, 2013.

[10] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data mining and knowledge discovery*, vol. 25, no. 1, pp. 1–33, 2012.

[11] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering on multi-layer graphs via subspace analysis on grassmann manifolds," *IEEE Transactions on signal processing*, vol. 62, no. 4, pp. 905–918, 2013.

[12] Z. Kuncheva and G. Montana, "Community detection in multiplex networks using locally adaptive random walks," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, 2015, pp. 1308–1315.

[13] M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall, "Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems," *Physical Review X*, vol. 5, no. 1, p. 011027, 2015.

[14] A. Amelio and C. Pizzuti, "Community detection in multidimensional networks," in *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*. IEEE, 2014, pp. 352–359.

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

[16] G. Zhu and K. Li, "A unified model for community detection of multiplex networks," in *International Conference on Web Information Systems Engineering*. Springer, 2014, pp. 31–46.

[17] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, ser. Proceedings, 2005-04-21, pp. 606–610.

[18] E. Cozzo, M. Kivelä, M. De Domenico, A. Solé-Ribalta, A. Arenas, S. Gómez, M. A. Porter, and Y. Moreno, "Structure of triadic relations in multiplex networks," *New Journal of Physics*, vol. 17, no. 7, p. 073029, 2015.

[19] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," vol. 13, 2001.

[20] W. Wu, S. Kwong, Y. Zhou, Y. Jia, and W. Gao, "Nonnegative matrix factorization with mixed hypergraph regularization for community detection," *Information Sciences*, vol. 435, pp. 263–281, 2018.

[21] H. Lu, X. Sang, Q. Zhao, and J. Lu, "Community detection algorithm based on nonnegative matrix factorization and pairwise constraints," *Physica A: Statistical Mechanics and its Applications*, vol. 545, p. 123491, 2020.

[22] 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, 2006, pp. 126–135.

[23] J. Yoo and S. Choi, "Orthogonal nonnegative matrix trifactorization for co-clustering: Multiplicative updates on stiefel manifolds," vol. 46, no. 5, pp. 559–570, 2010.

[24] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of statistical mechanics: Theory and experiment*, vol. 2005, no. 09, p. P09008, 2005.

[25] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen, "Quantitative function for community detection," *Physical review E*, vol. 77, no. 3, p. 036109, 2008.

[26] L. Jeub and M. Bazzi, "A generative model for mesoscale structure in multilayer networks implemented in matlab," 2016. [Online]. Available: https://github.com/MultilayerGM/MultilayerGM-MATLAB

[27] E. Lazega *et al.*, *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*. Oxford University Press on Demand, 2001.

[28] I. S. Jutla, L. G. Jeub, P. J. Mucha *et al.*, "A generalized louvain method for community detection implemented in matlab." [Online]. Available: https://github.com/GenLouvain/GenLouvain (2011-2019)