Discrete Overlapping Community Detection with Pseudo Supervision

Fanghua Ye*, Chuan Chen*, Zibin Zheng*, Rong-Hua Li†, Jeffrey Xu Yu‡
*School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
†School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
‡The Chinese University of Hong Kong, Hong Kong, China
smartyfh@outlook.com; {chenchuan, zhzibin}@mail.sysu.edu.cn; lironghuascut@gmail.com; yu@se.cuhk.edu.hk

***Abstract*—Community detection is of significant importance in understanding the structures and functions of networks. Recently, overlapping community detection has drawn much attention due to the ubiquity of overlapping community structures in real-world networks. Nonnegative matrix factorization (NMF), as an emerging standard framework, has been widely employed for overlapping community detection, which obtains nodes' soft community memberships by factorizing the adjacency matrix into low-rank factor matrices. However, in order to determine the ultimate community memberships, we have to post-process the real-valued factor matrix by manually specifying a threshold on it, which is undoubtedly a difficult task. Even worse, a unified threshold may not be suitable for all nodes. To circumvent the cumbersome post-processing step, we propose a novel discrete overlapping community detection approach, i.e., *Discrete Nonnegative Matrix Factorization (DNMF)*, which seeks for a discrete (binary) community membership matrix directly. Thus DNMF is able to assign explicit community memberships to nodes without post-processing. Moreover, DNMF incorporates a pseudo supervision module into it to exploit the discriminative information in an unsupervised manner, which further enhances its robustness. We thoroughly evaluate DNMF using both synthetic and real-world networks. Experiments show that DNMF has the ability to outperform state-of-the-art baseline approaches.**

***Index Terms*—community detection, overlapping communities, discrete nonnegative matrix factorization, pseudo supervision**

I. Introduction

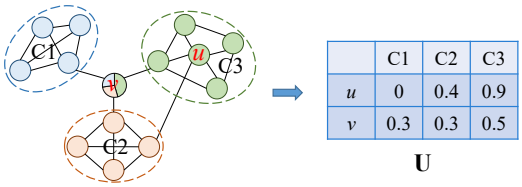
The research of network science has achieved rapid development in recent years. In fact, many real-world complex systems can be naturally characterized by the data structure of networks, such as social networks, information networks and biological neural networks [1]. One salient property of these networks is the existence of community structures. Intuitively, a community (also referred to as a module or a cluster) represents a group of cohesive nodes that have more connections inside the group than outside [1]. Admittedly, analyzing the underlying community structures is of significant importance in revealing the patterns and functions of networks. Besides, community detection has boosted plentiful applications, e.g., friend recommendation, team formation, semantic expansion and viral marketing [2].

Community detection has long been an important research topic in social network mining, web mining and social media analytics. Up to now, community detection has drawn enormous amounts of attention from various research fields [3]–[5], and extensive community detection approaches have been proposed [5]–[8]. However, the goal of traditional community detection approaches (e.g., Louvain [9] and Infomap [10]) is to partition a network into disjoint communities. Thus, each node is assigned to one and only one community, which is usually referred to as non-overlapping community detection. These traditional approaches ignore the fact that nodes participate naturally in multiple communities in real world. For example, in social networks, one individual can belong to more than one group such as colleague group, friend group and family group; in co-purchased networks, one item may also belong to multiple categories. Therefore, in order to learn the patterns and functions of networks adequately, it is necessary to uncover the overlapping community structures. This emerging issue is formally referred to as overlapping community detection [11].

Owing to the success of nonnegative matrix factorization (NMF) in machine learning [12], NMF has recently been adopted for community detection [13], [14]. Specifically, given an undirected and unweighted network with its adjacency matrix denoted by $\mathbf{A}$, NMF approximately factorizes $\mathbf{A}$ into two identical low-rank matrices $\mathbf{U}$ with nonnegative constraints, i.e., $\mathbf{A} \approx \mathbf{U}\mathbf{U}^T (\mathbf{U} \geq \mathbf{0})$. As thus, each column of $\mathbf{U}$ denotes a community and each row of $\mathbf{U}$ can be interpreted as the strength of associations between a node and different communities. In view of this, $\mathbf{U}$ is usually referred to as the community membership matrix. It is apparent that NMF fits into overlapping community detection. As a matter of fact, NMF has been regarded as a standard technique to deal with the problem of overlapping community detection, and various NMF-based approaches have been proposed [15]–[18].

In general, NMF-based overlapping community detection approaches consist of two stages: 1) factorizing the adjacency matrix $\mathbf{A}$ to obtain the community membership matrix $\mathbf{U}$ via optimizing certain objective function, and 2) determining the ultimate community assignments for each node via post-processing $\mathbf{U}$. The typical post-processing strategy is manually specifying a threshold on $\mathbf{U}$, then each entry of $\mathbf{U}$ that is larger than the threshold indicates a node-community assignment. Since $\mathbf{U}$ is a real-valued matrix, it is a nontrivial task to choose a proper threshold. An improper threshold may result in totally wrong community assignments. Moreover, a unified threshold may not be suitable for all nodes, and a personalized threshold for each node seems to be a better choice, which casts more difficulties on the post-processing. Take the network shown

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00081



(a) Network structure

	C1	C2	C3
u	0	0.4	0.9
v	0.3	0.3	0.5

U

(b) Community memberships

Fig. 1: (a) A toy network with three communities. Only node v joins in multiple communities. (b) The soft community memberships of nodes u and v. There is not a common threshold for nodes u and v.

in Fig. 1 as an example. Obviously, it is impossible to find a common threshold for nodes u and v on the given $\mathbf{U}$. For instance, if the threshold is set to be less than 0.3, then node u will be wrongly assigned to communities C_2 and C_3 simultaneously, although the community assignments of node v are correct. If the threshold is set to be larger than 0.3, then node v will not be assigned to communities C_1 and C_2 any more. Evidently, to obtain accurate community assignments, we need personalized thresholds for nodes u and v respectively (e.g., 0.8 for node u and 0.2 for node v). However, due to the lack of an oracle showing in advance how many communities each node belongs to, it is unrealistic to manually specify proper personalized thresholds for all nodes.

Targeting at skirting the cumbersome post-processing step, we propose a novel discrete overlapping community detection approach under the NMF framework. The proposed approach is named *Discrete Nonnegative Matrix Factorization (DNMF)*, which explicitly learns the discrete community memberships for each node. Therefore, there is no need to do the post-processing. More specifically, DNMF integrates the two stages of conventional NMF-based approaches into one, and it learns a binary community membership matrix $\mathbf{F}$ directly. Instead of restricting $\mathbf{U}$ to be binary, DNMF involves a smooth rotation matrix [19], [20] to transform the continuous $\mathbf{U}$ to the discrete $\mathbf{F}$. Thus, the continuous $\mathbf{U}$ just serves as an intermediate product. For ease of differentiation, we call $\mathbf{U}$ soft community membership matrix and $\mathbf{F}$ hard community membership matrix thereafter. To make DNMF more robust, we further incorporate a discriminative pseudo supervision module into it to learn a kernel regression [21] based prediction function by treating $\mathbf{F}$ as the ground-truth community labels and $\mathbf{A}$ as nodes' feature matrix. The rationality of introducing this module lies in that if $\mathbf{F}$ successfully captures the community memberships of nodes, then $\mathbf{F}$ should be predictable. It has been shown that such a pseudo supervision module is able to exploit the discriminative information in unsupervised scenarios [22]. Thus, DNMF is expected to have the capability of learning the intrinsic community memberships of nodes. Moreover, in DNMF, the community memberships and the pseudo supervision module are learnt in a unified manner with mutual guidance rather than separately, which further enhances the robustness of the learnt $\mathbf{F}$.

Overall, the main contributions of this paper include:

- To bypass the cumbersome post-processing step of conventional NMF-based overlapping community detection approaches, we propose DNMF to learn the discrete community memberships of nodes directly. To make DNMF more robust, we further incorporate a pseudo supervision module into it to exploit the discriminative information in an unsupervised manner.
- To address the discrete optimization of DNMF in a computationally tractable way, we devise an efficient learning algorithm which first decomposes the overall objective function into several independent subproblems by following the coordinate descent scheme [23], and then optimizes these subproblems alternately.
- To evaluate the performance of DNMF, we conduct extensive experiments on both synthetic and real-world networks. The results demonstrate the superiority of DNMF over state-of-the-art baseline approaches.

II. Related Work

During the past two decades, lots of efforts have been devoted to the research of community detection [5]. However, there is no universal definition for the community detection task. To design effective community detection models, a myriad of quantitative measurements have been proposed to assess the quality of community structures, e.g., modularity [24] and conductance [3]. Another way to model the community structures is to define explicit community models like k-core [25] and clique [26]. There are also some works resorting to clustering methods to identify communities [27]–[29].

However, most of the traditional research only focuses on non-overlapping community detection, which contradicts the fact that one node can naturally join in multiple communities. More recently, overlapping community detection has drawn much attention due to its relaxation to multiple community memberships [11]. For overlapping community detection, the classic approaches include clique percolation [30], link partitioning [31], label propagation [32], [33] and local expansion [34], [35]. The clique percolation method (CPM) is the first approach for overlapping community detection, and it is based on the assumption that each community is a union of adjacent k-cliques. Link partitioning reinvents communities as groups of links (edges) instead of nodes. Label propagation estimates the belonging coefficients of each node by averaging the coefficients of all its neighbor nodes. The local expansion methods are local-first approaches and they unveil communities from seeded small components.

Another line of research is based on NMF due to its high interpretability and its natural fitness for overlapping community detection. Psorakis et al. [16] first utilize a Bayesian NMF model to extract communities. Shi et al. [18] further propose an adaptive Bayesian NMF model for overlapping community detection. In [36], a bounded matrix tri-factorization method is proposed, where a third factor matrix is introduced to represent the interactions among all communities. Yang and Leskovec [15] develop an NMF-based generative model for community detection on massive networks, which relaxes the graph fitting problem into a continuous optimization problem. Zhang et al. [37] propose a preference-based NMF model to

TABLE I: Main symbols and parameters.

$\mathbf{A}$	The adjacency matrix
$\mathbf{U}$	The soft community membership matrix
$\mathbf{F}$	The hard community membership matrix
$\mathbf{Q}$	The rotation matrix
$\mathbf{S}$	The discrimination matrix
$\mathbf{K}$	The kernel matrix
$\hat{\mathbf{K}}$	The centered kernel matrix
$\mathbf{B}$	The kernel coefficient matrix
$\mathbf{b}$	The intercept vector
k	The number of communities
α	The tradeoff parameter
β	The tradeoff parameter
γ	The regularization parameter

incorporate implicit link preference information into overlapping community detection. Zhang et al. [17] further propose a homophily-based NMF approach to model the community-to-link and link-to-community effects simultaneously. Ye et al. [38] devise a deep autoencoder-like NMF model to learn the hierarchical mappings between the adjacency matrix $\mathbf{A}$ and the soft community membership matrix $\mathbf{U}$. He et al. [39] propose to identify and summarize communities concurrently under the NMF framework. For all these NMF-based approaches, a proper threshold should be manually specified on $\mathbf{U}$ so as to determine the ultimate community memberships.

Although there are different kinds of overlapping community detection methods as described above, NMF-based methods have gained more and more popularity in recent years. However, how to post-process the soft community membership matrix $\mathbf{U}$ to obtain accurate community assignments is an under-studied problem. Without reasonable post-processing, we may get poor community detection results even though $\mathbf{U}$ is of high quality. As aforementioned, it is troublesome to post-process $\mathbf{U}$ via manually specifying a threshold. In view of this, we propose to learn explicit community assignments directly without post-processing.

III. Preliminaries

Throughout this paper, we use bold uppercase and lowercase letters to denote matrices and vectors respectively. In particular, we use $\mathbf{X}_{ij}$ as the (i,j)-th entry of matrix $\mathbf{X}$. The i-th row and j-th column of $\mathbf{X}$ are denoted as $\mathbf{x}_i$ and $\mathbf{x}^j$. The Frobenius norm and trace of $\mathbf{X}$ are denoted as $\|\mathbf{X}\|_F$ and $tr(\mathbf{X})$. The Euclidean inner product between matrices $\mathbf{X}$ and $\mathbf{Y}$ is denoted as $< \mathbf{X}, \mathbf{Y} >$, i.e., $< \mathbf{X}, \mathbf{Y} >= tr(\mathbf{X}\mathbf{Y}^T)$. For a given vector $\mathbf{x}$, we denote $\|\mathbf{x}\|_2$ as its ℓ_2-norm. Besides, we denote $\mathbf{I}_n$ as the identity matrix of size $n \times n$ and $\mathbf{1}_n$ as an all-one column vector with n elements. The main symbols and parameters used in this paper are listed in Table I.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a network with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges, where $\mathcal{V}$ and $\mathcal{E}$ denote the node set and edge set respectively. In this paper, we focus on undirected and unweighted networks, thus we can represent $\mathcal{G}$ by its adjacency matrix $\mathbf{A} = (\mathbf{A}_{ij}) \in \{0,1\}^{n\times n}$ such that $\mathbf{A}_{ij} = 1$ if there is an edge between nodes i and j, and $\mathbf{A}_{ij} = 0$ otherwise. The target of community detection is to extract the underlying community structures of $\mathcal{G}$. Assume that $\mathcal{G}$ is composed of k communities. Let $\mathcal{C}$ denote the set of communities, i.e., $\mathcal{C} = \{C_i | C_i \neq \emptyset, C_i \neq C_j, 1 \leq i,j \leq k\}$, where C_i represents the i-th community. The traditional non-overlapping community detection problem requires that $C_i \cap C_j = \emptyset$ if $i \neq j$. However, in the overlapping community detection scenario, this constraint is removed.

Suppose that we have a soft community membership matrix $\mathbf{U} \in \mathbb{R}_+^{n\times k}$ with each entry $\mathbf{U}_{ij}$ representing the propensity of node i belonging to community C_j. Then, $\mathbf{U}_{ij}\mathbf{U}_{pj}$ can be treated as the expected number of edges between nodes i and p deduced by community C_j [40]. Summing over all the k communities, we obtain that the total expected number of edges between nodes i and p in network $\mathcal{G}$ is $\sum_{j=1}^{k} \mathbf{U}_{ij}\mathbf{U}_{pj}$. The expected number of edges implies that if two nodes share more similar community memberships, they have more possibilities to be linked. This generative process reflects the observation that nodes within communities are densely connected. Apparently, the expected number of edges between all pairs of nodes should be as closely consistent as possible with the adjacency matrix $\mathbf{A}$, i.e., $\mathbf{A} \approx \mathbf{U}\mathbf{U}^T$. There are various ways to measure the difference between $\mathbf{A}$ and $\mathbf{U}\mathbf{U}^T$. The most straightforward and widely-used method is in the form of $\|\mathbf{A} - \mathbf{U}\mathbf{U}^T\|_F^2$ [13].

IV. Discrete Overlapping Community Detection

To bypass the cumbersome post-processing step of conventional NMF-based overlapping community detection approaches, here we aim at seeking for a model that can learn the hard community membership matrix $\mathbf{F}$ directly. To achieve this goal, we propose the DNMF model for discrete overlapping community detection without post-processing.

A. Model Formulation

To explicitly learn the hard community membership matrix $\mathbf{F}$, we propose to introduce a rotation matrix $\mathbf{Q}$ to smoothly transform the continuous $\mathbf{U}$ to the discrete $\mathbf{F}$. We expect that $\mathbf{F}$ precisely captures the community structures and $\mathbf{F}$ is as closely consistent as possible with $\mathbf{U}$. To this end, we derive the following objective function:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{F},\mathbf{Q}} \|\mathbf{A} - \mathbf{U}\mathbf{U}^T\|_F^2 + \alpha\|\mathbf{U} - \mathbf{F}\mathbf{Q}\|_F^2, \\ &\text{s.t. } \mathbf{U} \geq \mathbf{0} \wedge \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k \wedge \mathbf{F} \in \mathcal{F}, \end{aligned} \tag{1}$$

where $\mathcal{F} = \{\mathbf{F} | \mathbf{F} \in \{0,1\}^{n\times k} \wedge \mathbf{F}\mathbf{1}_k \geq \mathbf{1}_n\}$ denotes the solution space of $\mathbf{F}$, $\mathbf{Q} \in \mathbb{R}^{k\times k}$ represents the rotation matrix and α is a trade-off parameter. The orthogonality constraint on $\mathbf{Q}$ is crucial, as it enforces $\mathbf{F}\mathbf{F}^T$ to be close to $\mathbf{U}\mathbf{U}^T$ (When minimizing (1), we have $\mathbf{U} \approx \mathbf{F}\mathbf{Q}$, then we can obtain $\mathbf{F}\mathbf{F}^T = \mathbf{F}\mathbf{Q}\mathbf{Q}^T\mathbf{F}^T = (\mathbf{F}\mathbf{Q})(\mathbf{F}\mathbf{Q})^T \approx \mathbf{U}\mathbf{U}^T$). In this regard, $\mathbf{F}$ preserves the information of $\mathbf{A}$ ($\mathbf{F}\mathbf{F}^T \approx \mathbf{U}\mathbf{U}^T \approx \mathbf{A}$), that is, $\mathbf{F}$ has the ability to capture the community structure information. Note that we do not directly replace $\mathbf{U}$ by $\mathbf{F}$ to learn nodes' hard community memberships for the reason that it will be highly intractable to solve the biquadratic discrete optimization problem [41]. However, with the aid of $\mathbf{Q}$, the learning of the model in (1) is much easier.

B. Discriminative Pseudo Supervision

It is noted that the model in (1) is in fact a generative model, as it seeks to reconstruct $\mathbf{A}$ from $\mathbf{U}$. Suppose that we are provided with the community label information, then we can learn a discriminative model directly in the supervised learning manner, e.g., by treating the columns of $\mathbf{A}$ as nodes' feature vectors, we can learn a regression-like model to predict the community labels of nodes readily. There is no doubt that the combination of generative and discriminative models can lead to better performance [42]. In view of this, we intend to extend the model in (1) to incorporate a discriminative supervision module into it. Considering that it is impractical to obtain the genuine overlapping community labels in most instances, we choose $\mathbf{F}$ as the pseudo ground-truth alternatively. As thus, the generative module and the discriminative module can be learnt in a mutual guidance manner, which empowers us to exploit discriminative information in an unsupervised manner.

To be more specific, given the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with its adjacency matrix denoted by $\mathbf{A} = (\mathbf{a}^1, \mathbf{a}^2, \cdots, \mathbf{a}^n)$, we treat $\mathbf{F}$ as the pseudo ground-truth community labels. Then, we can derive the following objective function to learn a robust kernel regression based prediction function:

$$\mathcal{L}(\mathbf{W}, \mathbf{b}; \mathbf{F}, \mathbf{A}) = \|\mathbf{F} - \phi^T(\mathbf{A})\mathbf{W} - \mathbf{1}_n\mathbf{b}^T\|_F^2 + \gamma\|\mathbf{W}\|_F^2, \quad (2)$$

where $\phi(\cdot)$ represents the kernel function and γ is a regularization parameter used to avoid overfitting.

Let $\mathbf{W} = \phi(\mathbf{A})\mathbf{B}$, where $\mathbf{B} \in \mathbb{R}^{n\times k}$ is defined as the kernel coefficient matrix, the above function is then equivalent to:

$$\begin{aligned}\mathcal{L}(\mathbf{B}, \mathbf{b}; \mathbf{F}, \mathbf{K}) =& \|\mathbf{F} - \phi^T(\mathbf{A})\phi(\mathbf{A})\mathbf{B} - \mathbf{1}_n\mathbf{b}^T\|_F^2 \\ &+ \gamma tr(\mathbf{B}^T\phi^T(\mathbf{A})\phi(\mathbf{A})\mathbf{B}) \\ =& \|\mathbf{F} - \mathbf{K}\mathbf{B} - \mathbf{1}_n\mathbf{b}^T\|_F^2 + \gamma tr(\mathbf{B}^T\mathbf{K}\mathbf{B}),\end{aligned} \quad (3)$$

where $\mathbf{K} = \phi^T(\mathbf{A})\phi(\mathbf{A})$ denotes the kernel matrix and its elements are given by $\mathbf{K}_{ij} = \phi^T(\mathbf{a}^i)\phi(\mathbf{a}^j)$. Up to now, lots of kernel functions have been proposed in the field of machine learning. In this paper, we choose the Gaussian kernel function [43] due to its high representation ability, thus the kernel matrix is computed as $\mathbf{K}_{ij} = \exp(-\|\mathbf{a}^i - \mathbf{a}^j\|_2^2/2\sigma^2)$, where σ denotes the kernel width. Here we set $\sigma = 1$ for simplicity.

Let $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ be a centering matrix, and let $\hat{\phi}(\mathbf{A}) = \phi(\mathbf{A})\mathbf{H}$, then the centered kernel matrix is computed as:

$$\hat{\mathbf{K}} = \hat{\phi}^T(\mathbf{A})\hat{\phi}(\mathbf{A}) = \mathbf{H}^T\phi^T(\mathbf{A})\phi(\mathbf{A})\mathbf{H} = \mathbf{H}^T\mathbf{K}\mathbf{H}. \quad (4)$$

By replacing the kernel matrix $\mathbf{K}$ with the centered kernel matrix $\hat{\mathbf{K}}$ in (3), we arrive at:

$$\mathcal{L}(\mathbf{B}, \mathbf{b}; \mathbf{F}, \hat{\mathbf{K}}) = \|\mathbf{F} - \hat{\mathbf{K}}\mathbf{B} - \mathbf{1}_n\mathbf{b}^T\|_F^2 + \gamma tr(\mathbf{B}^T\hat{\mathbf{K}}\mathbf{B}). \quad (5)$$

When $\mathbf{F}$ and $\hat{\mathbf{K}}$ are given, the minimization of (5) can be solved by setting the derivatives of $\mathcal{L}(\mathbf{B}, \mathbf{b}; \mathbf{F}, \hat{\mathbf{K}})$ with respect to $\mathbf{B}$ and $\mathbf{b}$ to 0. Then, we obtain:

$$\mathbf{B} = (\hat{\mathbf{K}} + \gamma\mathbf{I}_n)^{-1}\mathbf{F}, \quad \mathbf{b} = \frac{1}{n}\mathbf{F}^T\mathbf{1}_n. \quad (6)$$

By substituting (6) into (5), we have:

$$\mathcal{L}(\mathbf{B}, \mathbf{b}; \mathbf{F}, \hat{\mathbf{K}}) = tr(\mathbf{F}^T\mathbf{S}\mathbf{F}), \quad (7)$$

where $\mathbf{S} = \mathbf{H} - (\hat{\mathbf{K}} + \gamma\mathbf{I}_n)^{-1}\hat{\mathbf{K}}$. For convenience, we call $\mathbf{S}$ the discrimination matrix.

C. The Unified Model

Our DNMF model chooses to learn the hard community membership matrix $\mathbf{F}$ and the discriminative pseudo supervision module in a unified manner rather than separately. To this end, we combine (1) and (7) together and obtain the final objective function as follows:

$$\begin{aligned}&\min_{\mathbf{U},\mathbf{F},\mathbf{Q}} \|\mathbf{A} - \mathbf{U}\mathbf{U}^T\|_F^2 + \alpha\|\mathbf{U} - \mathbf{F}\mathbf{Q}\|_F^2 + \beta\mathcal{L}(\mathbf{B}, \mathbf{b}; \mathbf{F}, \hat{\mathbf{K}}) \\ &\quad = \|\mathbf{A} - \mathbf{U}\mathbf{U}^T\|_F^2 + \alpha\|\mathbf{U} - \mathbf{F}\mathbf{Q}\|_F^2 + \beta tr(\mathbf{F}^T\mathbf{S}\mathbf{F}), \\ &\quad \text{s.t. } \mathbf{U} \geq \mathbf{0} \wedge \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k \wedge \mathbf{F} \in \mathcal{F},\end{aligned} \quad (8)$$

where β is a trade-off parameter.

V. Optimization

The objective function in (8) is not convex over the three variables $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{Q}$ simultaneously. To solve it, we propose an efficient learning algorithm by following the coordinate descent scheme [23]. First, (8) is decomposed into three subproblems with respect to $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{Q}$ respectively. Then, (8) is efficiently optimized by solving the three subproblems alternately.

A. Alternating Optimization

1) The $\mathbf{U}$*-subproblem:* When $\mathbf{F}$ and $\mathbf{Q}$ are fixed, we need to solve the following $\mathbf{U}$-subproblem:

$$\min_{\mathbf{U}\geq\mathbf{0}} \|\mathbf{A} - \mathbf{U}\mathbf{U}^T\|_F^2 + \alpha\|\mathbf{U} - \mathbf{F}\mathbf{Q}\|_F^2, \quad (9)$$

which can be further rewritten as follows:

$$\begin{aligned}\min_{\mathbf{U}\geq\mathbf{0}}\ & tr(\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{U}^T) - 2tr(\mathbf{A}\mathbf{U}\mathbf{U}^T) + \alpha tr(\mathbf{U}\mathbf{U}^T) \\ & - 2\alpha tr(\mathbf{F}\mathbf{Q}^+\mathbf{U}^T) + 2\alpha tr(\mathbf{F}\mathbf{Q}^-\mathbf{U}^T),\end{aligned} \quad (10)$$

where the terms irrelevant to $\mathbf{U}$ are omitted and matrices $\mathbf{Q}^+$ and $\mathbf{Q}^-$ denote the positive and negative parts of $\mathbf{Q}$ respectively. That is,

$$\mathbf{Q}^+ = \frac{|\mathbf{Q}| + \mathbf{Q}}{2}, \quad \mathbf{Q}^- = \frac{|\mathbf{Q}| - \mathbf{Q}}{2},$$

where $|\mathbf{Q}|$ represents the absolute value of $\mathbf{Q}$.

Although the objective function in (10) does not have a closed-form solution, it can be optimized by iteratively updating $\mathbf{U}$ according to Theorem 1.

Theorem 1. *While fixing* $\mathbf{F}$ *and* $\mathbf{Q}$*, updating* $\mathbf{U}$ *according to* (11) *will monotonically decrease the objective function in* (10) *until convergence. At convergence, the solution is a KKT fixed point.*

$$\mathbf{U} \leftarrow \mathbf{U} \odot \left(\frac{2\mathbf{A}\mathbf{U} + \alpha\mathbf{F}\mathbf{Q}^+}{2\mathbf{U}\mathbf{U}^T\mathbf{U} + \alpha\mathbf{U} + \alpha\mathbf{F}\mathbf{Q}^-}\right)^{\frac{1}{4}}. \quad (11)$$

The proof of Theorem 1 can be found in the appendix. Note that in (11), $\odot$, $\frac{[\cdot]}{[\cdot]}$ and $([\cdot])^{\frac{1}{4}}$ are all element-wise operators. On the basis of Theorem 1, the optimization process of the $\mathbf{U}$-subproblem is summarized in Algorithm 1.

Algorithm 1 Algorithm for solving the $\mathbf{U}$-subproblem.

Input: The adjacency matrix $\mathbf{A}$, the initial matrix $\mathbf{U}_0$, the fixed matrices $\mathbf{F}$, $\mathbf{Q}$, and parameter α;
Output: The soft community membership matrix $\mathbf{U}$;
1: Initialize $\mathbf{U} = \mathbf{U}_0$;
2: **while** not converged **do**
3: Update $\mathbf{U}$ according to (11);
4: **end while**
5: **return** $\mathbf{U}$;

Algorithm 2 Algorithm for solving the $\mathbf{F}$-subproblem.

Input: The discrimination matrix $\mathbf{S}$, the initial matrix $\mathbf{F}_0$, the fixed matrices $\mathbf{U}$, $\mathbf{Q}$, and parameters α, β;
Output: The hard community membership matrix $\mathbf{F}$;
1: Initialize $\mathbf{F} = \mathbf{F}_0$;
2: **repeat**
3: **for** $i = 1$ **to** n **do**
4: Update $\mathbf{f}_i$ according to (15);
5: **end for**
6: **until** there is no change to $\mathbf{F}$
7: **return** $\mathbf{F}$;

2) The $\mathbf{F}$-subproblem: When $\mathbf{U}$ and $\mathbf{Q}$ are fixed, we need to solve the following $\mathbf{F}$-subproblem:

$$\min_{\mathbf{F}} \ \alpha\|\mathbf{U} - \mathbf{F}\mathbf{Q}\|_F^2 + \beta tr(\mathbf{F}^T\mathbf{S}\mathbf{F}), \ \text{s.t.} \ \mathbf{F} \in \mathcal{F}, \tag{12}$$

which is equivalent to the following objective function:

$$\min_{\mathbf{F}} \ tr(\mathbf{F}^T\mathbf{S}'\mathbf{F}) - 2\alpha tr(\mathbf{U}\mathbf{Q}^T\mathbf{F}^T), \ \text{s.t.} \ \mathbf{F} \in \mathcal{F}, \tag{13}$$

where $\mathbf{S}' = \beta\mathbf{S} + \alpha\mathbf{I}_n$. Due to the discrete constraints, the above minimization problem is generally NP-hard. We choose to update the hard community membership matrix $\mathbf{F}$ by using the *discrete coordinate descent (DCD)* method [44].

Let $\mathbf{f}_i$ denote the i-th row of $\mathbf{F}$ and $\mathbf{F}'$ the matrix of $\mathbf{F}$ excluding $\mathbf{f}_i$. Further, let $\mathbf{s}_i'$ denote the i-th row of $\mathbf{S}'$ while the element $\mathbf{S}'_{ii}$ is excluded, and let $\mathbf{q}_i$ denote the i-th row of $\mathbf{U}\mathbf{Q}^T$. DCD will update the i-th row $\mathbf{f}_i$ while fixing all the other rows of $\mathbf{F}$. Then, we have $tr(\mathbf{F}^T\mathbf{S}'\mathbf{F}) = const + \mathbf{S}'_{ii}\mathbf{f}_i\mathbf{f}_i^T + 2\mathbf{s}_i'\mathbf{F}'\mathbf{f}_i^T$ and $tr(\mathbf{U}\mathbf{Q}^T\mathbf{F}^T) = const + \mathbf{q}_i\mathbf{f}_i^T$. Putting the two equations together, we derive the following optimization problem with respect to $\mathbf{f}_i$:

$$\min_{\mathbf{f}_i} \mathbf{S}'_{ii}\mathbf{f}_i\mathbf{f}_i^T + 2(\mathbf{s}_i'\mathbf{F}' - \alpha\mathbf{q}_i)\mathbf{f}_i^T, \text{s.t.} \ \mathbf{f}_i \in \{0,1\}^k \wedge \mathbf{f}_i\mathbf{1}_k \geq 1. \tag{14}$$

Let $\mathbf{e} = \mathbf{S}'_{ii}\mathbf{1}_k^T + 2(\mathbf{s}_i'\mathbf{F}' - \alpha\mathbf{q}_i)$ with its j-th entry denoted by $\mathbf{e}_j$. Then, the above problem has the optimal solution as:

$$(\mathbf{f}_i)_j = \begin{cases} 1, & \text{if } \mathbf{e}_j = \min(\mathbf{e}) \vee \mathbf{e}_j < 0, \\ 0, & \text{if } \mathbf{e}_j \neq \min(\mathbf{e}) \wedge \mathbf{e}_j \geq 0. \end{cases} \tag{15}$$

The optimization process of the $\mathbf{F}$-subproblem is summarized in Algorithm 2.

Algorithm 3 Algorithm for DNMF.

Input: The adjacency matrix $\mathbf{A}$, and parameters k, α, β, γ;
Output: $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{Q}$;
1: Initialize $\mathbf{U}$, $\mathbf{F}$, $\mathbf{Q}$;
2: **while** not converged **do**
3: Solve the $\mathbf{U}$-subproblem by invoking Algorithm 1;
4: Solve the $\mathbf{F}$-subproblem by invoking Algorithm 2;
5: Solve the $\mathbf{Q}$-subproblem according to (17);
6: **end while**
7: **return** $\mathbf{U}$, $\mathbf{F}$, $\mathbf{Q}$;

3) The $\mathbf{Q}$-subproblem: When $\mathbf{U}$ and $\mathbf{F}$ are fixed, we need to solve the following $\mathbf{Q}$-subproblem:

$$\min_{\mathbf{Q}} \ \|\mathbf{U} - \mathbf{F}\mathbf{Q}\|_F^2, \ \text{s.t.} \ \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k. \tag{16}$$

The optimal solution of $\mathbf{Q}$ is given by Theorem 2.

Theorem 2. *With $\mathbf{U}$ and $\mathbf{F}$ fixed, the closed-form solution of* (16) *is given as follows:*

$$\mathbf{Q} = \mathbf{\Omega}_2{\mathbf{\Omega}_1}^T, \tag{17}$$

where $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are the left and right singular vectors of the Singular Value Decomposition (SVD) of $\mathbf{U}^T\mathbf{F}$, i.e., $\mathbf{U}^T\mathbf{F} = \mathbf{\Omega}_1\mathbf{\Sigma}{\mathbf{\Omega}_2}^T$.

Proof. Note that $\|\mathbf{U} - \mathbf{F}\mathbf{Q}\|_F^2 = tr(\mathbf{U}\mathbf{U}^T - 2\mathbf{U}^T\mathbf{F}\mathbf{Q} + \mathbf{F}\mathbf{Q}\mathbf{Q}^T\mathbf{F}^T)$. Considering the fact that $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_k$, we can then obtain: $\min_{\mathbf{Q}}\|\mathbf{U} - \mathbf{F}\mathbf{Q}\|_F^2 \Leftrightarrow \max_{\mathbf{Q}} tr(\mathbf{U}^T\mathbf{F}\mathbf{Q}) \Leftrightarrow \max_{\mathbf{Q}} tr(\mathbf{\Omega}_1\mathbf{\Sigma}{\mathbf{\Omega}_2}^T\mathbf{Q}) \Leftrightarrow \max_{\mathbf{Q}} < \mathbf{\Omega}_1\sqrt{\mathbf{\Sigma}}, \mathbf{Q}^T\mathbf{\Omega}_2\sqrt{\mathbf{\Sigma}} >$. According to the Cauchy-Schwarz inequality, we have:

$$< \mathbf{\Omega}_1\sqrt{\mathbf{\Sigma}}, \mathbf{Q}^T\mathbf{\Omega}_2\sqrt{\mathbf{\Sigma}} > \ \leq \ \|\mathbf{\Omega}_1\sqrt{\mathbf{\Sigma}}\|_F\|\mathbf{Q}^T\mathbf{\Omega}_2\sqrt{\mathbf{\Sigma}}\|_F.$$

The equality holds when $\mathbf{\Omega}_1\sqrt{\mathbf{\Sigma}} = \mathbf{Q}^T\mathbf{\Omega}_2\sqrt{\mathbf{\Sigma}}$. Therefore, we obtain the optimal solution $\mathbf{Q} = \mathbf{\Omega}_2{\mathbf{\Omega}_1}^T$. □

By now we have presented the solution for each subproblem. The whole optimization process is then summarized in Algorithm 3. By optimizing the three subproblems alternately, the learning algorithm is able to decrease the objective function in (8) monotonically in each iteration until convergence.

B. Time Complexity

The main time cost of Algorithm 3 lies in the solving of the three subproblems. Thus, we analyze the time complexity for each subproblem respectively. For the $\mathbf{U}$-subproblem, it takes $\mathcal{O}(t_u(n^2k + nk^2))$ time, where t_u denotes the number of iterations to achieve convergence. For the $\mathbf{F}$-subproblem, it takes $\mathcal{O}(nk)$ time for completing the inner loop of updating $\mathbf{f}_i$, thus the overall time complexity of the $\mathbf{F}$-subproblem is $\mathcal{O}(t_f n^2 k)$, where t_f denotes the number of iterations for convergence. For the $\mathbf{Q}$-subproblem, it takes $\mathcal{O}(nk^2 + k^3)$ time, which can be simplified as $\mathcal{O}(nk^2)$ due to $k \ll n$. Assume that Algorithm 3 requires t_t iterations for convergence, then its overall time complexity is of order $\mathcal{O}(t_t((t_u + t_f)n^2k + t_u nk^2))$. As can be seen, although we have imposed strict discrete constraints on $\mathbf{F}$, the proposed learning algorithm is efficient to optimize the objective function.

VI. Experiments

In this section, we conduct extensive experiments on both synthetic and real-world networks to evaluate the performance of our DNMF model. The learning algorithm is implemented by MATLAB 2016b and all experiments are conducted on a server with two 2.4GHz Intel Xeon CPUs and 128GB main memory running Ubuntu 14.04.5 (64-bit). The source code is available at https://github.com/smartyfh/DNMF.

A. Baseline Methods

We select seven state-of-the-art overlapping community detection approaches as baseline methods of our DNMF model.

BigClam: BigClam is a cluster affiliation model [15]. BigClam relaxes the graph fitting problem into a continuous optimization problem, which makes it suitable for dealing with large-scale networks.

DEMON: DEMON is a local-first approach for overlapping community detection [45]. DEMON unveils the community structures by letting each node vote for the communities it sees surrounding it in its ego neighborhood.

HNMF: HNMF is a probabilistic approach [17]. HNMF seeks to model the link-to-community and community-to-link perspectives simultaneously based on the assumptions that nodes are more likely to build links if they share communities and that linked nodes are more similar than non-linked nodes.

EgoSplit: EgoSplit is a highly scalable and flexible framework for detecting overlapping communities [6]. EgoSplit works in two steps: a local ego-net analysis phase and a global graph partitioning phase. In the global graph partitioning phase, the traditional Louvain algorithm [9] is used.

NSED: NSED is a nonnegative symmetric encoder-decoder approach proposed for community detection [46]. Similar to autoencoder, NSED involves an encoder component and a decoder component concurrently.

OCDDP: OCDDP is an overlapping community detection algorithm based on density peaks [47]. OCDDP utilizes a similarity-based method to set distances among nodes, a three-step process to select cores of communities and membership vectors to represent belongings of nodes.

LFCIS: LFCIS is a latent factor model, which takes both network topology and node features into consideration [39]. Similar to DNMF, we treat each column of the adjacency matrix $\mathbf{A}$ as nodes' feature vectors for LFCIS.

Among these baseline methods, DEMON, EgoSplit and OCDDP are able to determine the number of underlying communities automatically, while the number of communities to detect for BigClam, HNMF, NSED and LFCIS should be pre-determined. Besides, for methods BigClam, HNMF, NSED and LFCIS, after obtaining the soft community memberships, a proper threshold should be manually specified to identify the hard community memberships. We take the same method as in [15] to determine the threshold. Then the threshold is calculated as $\sqrt{-\log(1-\frac{2m}{n(n-1)})}$, where n and m denote the number of nodes and the number of edges respectively. For a fair comparison, we run each method 10 times and the average results are reported.

B. Evaluation Metrics

For networks having ground-truth communities, we choose the overlapping normalized mutual information (**ONMI**) [35] as the evaluation metric. ONMI estimates the similarity between the ground-truth community memberships $\mathcal{C}^*$ and the detected ones $\mathcal{C}$. Formally, ONMI is calculated as below:

$$ONMI = 1 - \frac{1}{2}\left(\sum_{i}^{|\mathcal{C}|}\frac{H(C_i|\mathcal{C}^*)}{|\mathcal{C}|H(C_i)} + \sum_{j}^{|\mathcal{C}^*|}\frac{H(C_j^*|\mathcal{C})}{|\mathcal{C}^*|H(C_j^*)}\right),$$

where $H(C_i)$ denotes the entropy of the i-th community C_i, and $H(C_i|\mathcal{C}^*)$ denotes the entropy of C_i with respect to $\mathcal{C}^*$. $H(C_i|\mathcal{C}^*)$ is given by:

$$H(C_i|\mathcal{C}^*) = \min_{l\in\{1,2,\cdots,|\mathcal{C}^*|\}} H(C_i|C_l^*),$$

where $H(C_i|C_l^*)$ denotes the conditional entropy of C_i on C_l^*.

For networks without ground-truth communities, we choose the most widely adopted **Modularity** [24] as the evaluation metric. The original modularity is defined for evaluating non-overlapping communities. Following [48], here we slightly modify the definition to make it suitable for quantifying overlapping communities. The modified modularity Q is defined as follows:

$$Q = \frac{1}{2m}\sum_{i=1}^{|\mathcal{C}|}\sum_{u\in C_i, v\in C_i}\frac{1}{O_u O_v}\left(\mathbf{A}_{uv} - \frac{\kappa_u\kappa_v}{2m}\right),$$

where m denotes the number of edges, O_u denotes the number of communities that node u belongs to and κ_u denotes the degree of node u.

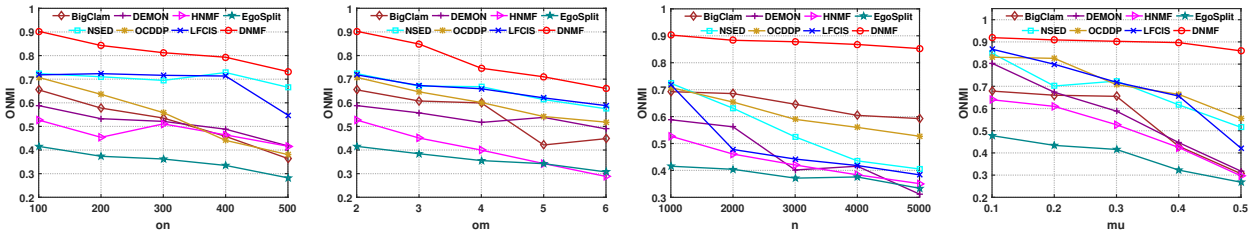
For both evaluation metrics, higher values indicate more accurately detected communities, i.e., the detected community memberships correspond better to the ground-truth ones.

C. Experiments on Synthetic Networks

1) Datasets: We first conduct experiments on synthetic networks. We employ the well-known LFR toolkit [35] to generate synthetic networks with ground-truth overlapping community structures. The parameters of the LFR benchmarks are listed in Table II. We have generated a total of 20 LFR benchmark networks by varying the number of nodes n, the number of overlapping nodes on, the number of memberships of the overlapping nodes om and the mixing parameter mu (each node shares a fraction of its edges with nodes in other communities). All the other parameters are set fixedly as shown in Table II.

2) Experimental Results: In the experiments, for all baseline methods, we tune the corresponding parameters by following the guidance of their authors. For our method DNMF, we tune α in the range of $\{0.01, 0.05, 0.1, 0.5, 1, 5\}$, both β and γ in the range of $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$. For methods BigClam, HNMF, NSED, LFCIS and DNMF, we set the number of communities (i.e., k) to be the ground-truth to test whether the genuine communities can be identified or not.

We first generate 5 networks by varying on from 100 to 500 with a step size of 100 and fixing n at 1000, om at 2 and



(a) Networks with different on (b) Networks with different om (c) Networks with different n (d) Networks with different mu

Fig. 2: Performance comparison in terms of ONMI on twenty LFR benchmark networks.

TABLE II: Parameters of LFR benchmark networks.

n	Number of nodes	To vary
on	Number of overlapping nodes	To vary
om	Number of overlapping memberships	To vary
mu	Mixing parameter	To vary
d_{avg}	Average degree	Fixed at 20
d_{max}	Maximum degree	Fixed at 50
t_1	Exponent for node degree distribution	Fixed at 2
t_2	Exponent for community size distribution	Fixed at 1
C_{min}	Minimum community size	Fixed at $\frac{n}{50}$
C_{max}	Maximum community size	Fixed at $\frac{n}{10}$

mu at 0.3. The results on these networks are shown in Fig. 2 (a). As can be seen, our DNMF model consistently shows better performance than all baseline methods. We can also see that EgoSplit shows the worst performance, this is because it identifies far more communities than the ground-truth.

We then generate 5 networks by varying om from 2 to 6 with a step size of 1 and fixing n at 1000, on at 100 and mu at 0.3. The results on these networks are shown in Fig. 2 (b), where similar results as before can be observed. Furthermore, from both Fig. 2 (a) and Fig. 2 (b), it can be observed that when on or om grows larger, the performance of all methods tends to decrease. This is because when more nodes have overlapping community memberships or overlapping nodes participate in more communities, the community detection task will become more challenging.

Next, we generate 5 networks by varying n from 1000 to 5000 and fixing on at $\frac{n}{10}$, om at 2 and mu at 0.3. We further generate 5 networks by varying mu from 0.1 to 0.5 and fixing n at 1000, on at 100 and om at 2. The results on these networks are illustrated in Fig. 2 (c) and Fig. 2 (d) respectively. On these two group of networks, DNMF shows significant better performance than the other methods. Even on networks with a large mu which no longer have distinct community structures, our model still shows satisfactory performance. These results demonstrate that our model indeed has the ability to detect discrete overlapping communities without the cumbersome post-processing.

3) Node Level Analysis: Although previous experiments have proven the superior performance of our model, they cannot provide us with an intuitive understanding of the superiority of DNMF. In this part, we further analyze the results from node-level perspective. Apparently, the most useful information associated with each node is the number of communities it belongs to. Therefore, we look at the number

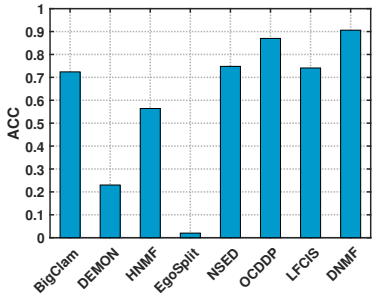


Fig. 3: Node-level analysis.

TABLE III: Statistics of real-world networks. n: number of nodes, m: number of edges, and k: pre-determined number of communities.

Networks	n	m	k
Dolphins	62	159	5
Football	115	613	10
Jazz	198	2742	5
Metabolic	453	2025	20
Netscience	1589	2742	35
Powergrid	4941	6594	45
Pubmed	19717	44338	5

of detected community memberships relative to the ground-truth one, and we define a new evaluation metric **Accuracy**, which is calculated as follows:

$$ACC = \frac{1}{n}\sum_{u\in\mathcal{V}} \mathbf{1}\{O_u = O_u^*\},$$

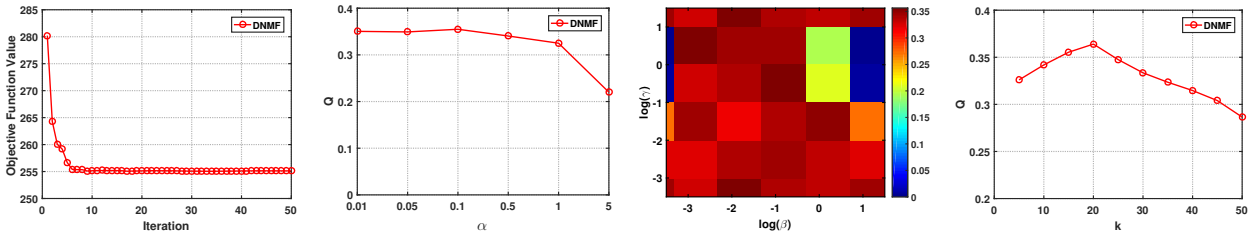
where O_u^* denotes the number of ground-truth communities that node u belongs to. Clearly, higher ACC values indicate that there are more nodes sharing the same number of community memberships with the ground-truth. We report the results on the network generated by fixing n at 1000, on at 100, om at 2 and mu at 0.3 in Fig. 3. It can be seen that DNMF reaches the highest ACC value. The results intuitively show why our model can achieve better performance.

D. Experiments on Real-World Networks

1) Datasets: In this part, we conduct experiments on seven real-world networks. These networks are downloaded from (http://www-personal.umich.edu/~mejn/netdata/ and https://linqs.soe.ucsc.edu/). Their basic information is listed in Table III. Since all these networks have no ground-truth overlapping communities, we use modularity Q as the evaluation metric. In addition, we set the number of communities to detect (i.e., k) for BigClam, HNMF, NSED, LFCIS and DNMF as

TABLE IV: Performance comparison in terms of modularity Q (mean $\pm$ standard deviation).

Networks	BigClam	DEMON	HNMF	EgoSplit	NSED	OCDDP	LFCIS	DNMF
Dolphins	0.365±0.038	0.319±0.004	0.342±0.031	0.194±0.006	0.374±0.039	0.486±0.011	0.393±0.002	**0.524**±0.009
Football	0.491±0.039	0.398±0.007	0.484±0.023	0.128±0.002	0.553±0.033	0.483±0.076	0.589±0.002	**0.601**±0.004
Jazz	0.342±0.029	0.071±0.010	0.342±0.021	0.383±0.075	0.395±0.005	0.261±0.026	0.396±0.002	**0.423**±0.003
Metabolic	0.194±0.016	0.064±0.002	0.156±0.002	0.336±0.057	0.108±0.009	0.041±0.008	0.087±0.008	**0.357**±0.008
Netscience	0.589±0.005	0.611±0.007	0.446±0.070	0.769±0.002	0.519±0.037	0.732±0.125	0.343±0.008	**0.904**±0.010
Powergrid	0.484±0.004	0.083±0.001	0.129±0.008	0.188±0.001	0.445±0.025	**0.794**±0.108	0.001±0.103	0.789±0.004
Pubmed	0.037±0.006	0.002±0.001	0.345±0.015	0.055±0.001	0.249±0.064	0.501±0.018	0.003±0.001	**0.566**±0.027



(a) Convergence analysis (b) Effects of α (c) Joint effects of β and γ (d) Effects of k

Fig. 4: Convergence and parameters sensitivity analyses of DNMF.

follows: we run symmetric NMF [13] on each network by varying k from 5 to 50 with a step size of 5, and then the k that achieves the highest Q is selected. The detailed information is shown in Table III.

2) Experimental Results: The results in terms of Q are presented in Table IV, where the best results are presented in bold numbers. Table IV shows that DNMF significantly outperforms the baseline methods on all networks, except for performing the second best on Powergrid. The results verify again that DNMF indeed has the ability to capture the discrete community memberships of nodes accurately.

3) Convergence and Parameters Sensitivity Analyses: Here, we empirically analyze the convergence and parameters sensitivity of DNMF on Metabolic. Similar results can be observed on other networks.

The results about convergence are shown in Fig. 4 (a), from which, we can see that DNMF converges rapidly. The objective function value becomes stable within only a few iterations. The results confirm that our learning algorithm (i.e., Algorithm 3) is quite efficient.

The results with respect to parameter α are shown in Fig. 4 (b). As can be seen, DNMF is robust to α when α is small. However, when α is large, the performance degrades severely. Recall that α plays a significant role in learning the hard community memberships. The results manifest that our model is very effective in learning the discrete overlapping community structures when α does not take too large values.

We further report the joint effects of parameters β and γ in Fig. 4 (c). Recall that β and γ control the contribution of the pseudo supervision module. It can be observed that DNMF is robust to β and γ. As long as β and γ do not take too large values simultaneously, DNMF is able to achieve satisfactory performance.

We also show the effects of the number of communities on the performance of DNMF by varying k from 5 to 50 with a step size of 5. The results are illustrated in Fig. 4 (d). It shows that when k takes value the same as the pre-determined one (i.e., 20), DNMF shows the best performance. The results demonstrate that although DNMF needs to specify the number of communities to detect, it has the capability to determine which k gives the best performance. Thus, we can use binary search to find the most appropriate k in practice.

4) Ablation Study: Recall that DNMF involves a pseudo supervision module to exploit discriminative information. Here, we conduct an ablation study to test whether the pseudo supervision module is able to improve the performance of DNMF. We denote **DNMFp** as the pruned version of DNMF without the pseudo supervision module. We also compare DNMF with **SNMF** (short for symmetric NMF [13]), which is the fundamental component of DNMF (when $\alpha = 0$ and $\beta = 0$). The results are shown in Fig. 5. As can be seen, both DNMF and DNMFp outperform SNMF on all networks. DNMF also consistently shows better performance than DNMFp. The results verify that the pseudo supervision module can lead to better performance.

E. Running Time Analysis

1) Running Time Comparison: In this subsection, we compare the running time of different methods on a synthetic LFR network with 5000 nodes. The results are illustrated in Fig. 6. As can be seen, DNMF shows comparable time overhead with HNMF. It runs even faster than BigClam. The results confirm that our learning algorithm is very efficient, in spite of the discrete constraints.

2) Scalability Testing: We further report the scalability of DNMF. To this end, we generate nine LFR benchmark networks by varying the number of nodes from 10000 to 50000 with a step size of 5000. In this experiment, the number of iterations of the learning algorithm is fixed at 50 for all networks. The results are shown in Fig. 7. As can be seen, when $n = 10000$, DNMF can finish the learning process in several minutes. With the increasing of n, the time cost grows near linearly. The results verify the efficiency of our learning algorithm again. Note that we do not test the scalability of

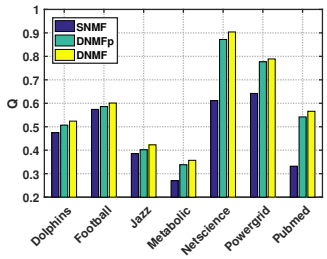


Fig. 5: Ablation study.

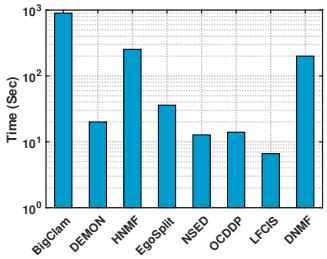


Fig. 6: Running time comparison.

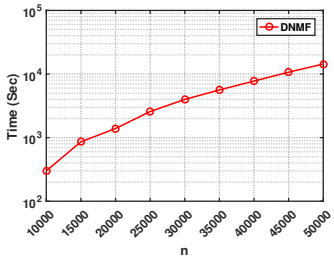


Fig. 7: Scalability testing.

DNMF on further larger networks for the reason that their adjacency matrices cannot be loaded into the main memory of a single machine. Therefore, it is valuable to implement DNMF in a distributed way to deal with super-large scale networks, which we leave as our future work.

VII. Conclusion

In this paper, we have proposed a novel overlapping community detection model DNMF, which can learn the discrete community memberships of nodes directly, thus there is no need to do the thorny post-processing. We have further incorporated a pseudo supervision module into DNMF to exploit the discriminative information in an unsupervised manner. The mutual guidance between the learning of the community memberships and the learning of the pseudo supervision module can strengthen the robustness of DNMF. To solve DNMF effectively, we have introduced an efficient learning algorithm with detailed convergence analysis. The experiments on both synthetic and real-world networks have verified the efficiency and effectiveness of DNMF comprehensively.

Acknowledgement

The work described in this paper was supported by the National Key Research and Development Program (2016YFB1000101), the National Natural Science Foundation of China (11801595, 61772346, U1809206), the Research Grants Council of Hong Kong, China (14203618), the Natural Science Foundation of Guangdong (2018A030310076) and the CCF-Tencent Open Fund WeBank Special Funding. Chuan Chen is the corresponding author.

Appendix

We first show that the limited solution of the update rule in (11) satisfies the KKT condition. To this end, we introduce the Lagrangian function:

$$\mathcal{L}(\mathbf{U}) = \|\mathbf{A} - \mathbf{U}\mathbf{U}^T\|_F^2 + \alpha\|\mathbf{U} - \mathbf{F}\mathbf{Q}\|_F^2 - tr(\mathbf{\Theta}\mathbf{U}^T), \quad (18)$$

where the Lagrangian multiplier matrix $\mathbf{\Theta}$ enforces nonnegative constraints, $\mathbf{U} \geq 0$. The zero gradient condition gives $\partial\mathcal{L}(\mathbf{U})/\partial\mathbf{U} = 4(\mathbf{U}\mathbf{U}^T - \mathbf{A})\mathbf{U} + 2\alpha(\mathbf{U} - \mathbf{F}\mathbf{Q}) - \mathbf{\Theta} = 0$. According to the complementary slackness condition $\mathbf{\Theta}_{ij}\mathbf{U}_{ij} = 0$, we obtain:

$$\big(2(\mathbf{U}\mathbf{U}^T - \mathbf{A})\mathbf{U} + \alpha(\mathbf{U} - \mathbf{F}\mathbf{Q}^+ + \mathbf{F}\mathbf{Q}^-)\big)_{ij}\mathbf{U}_{ij} = 0, \quad (19)$$

which is a fixed point equation that the solution must satisfy at convergence.

It is easy to show that the limited solution of the update rule in (11) satisfies this fixed point equation. At convergence, we have $\mathbf{U}^{(\infty)} = \mathbf{U}^{(t+1)} = \mathbf{U}^{(t)} = \mathbf{U}$, that is,

$$\mathbf{U}_{ij} = \mathbf{U}_{ij}\left(\frac{2(\mathbf{A}\mathbf{U})_{ij} + \alpha(\mathbf{F}\mathbf{Q}^+)_{ij}}{2(\mathbf{U}\mathbf{U}^T\mathbf{U})_{ij} + \alpha\mathbf{U}_{ij} + \alpha(\mathbf{F}\mathbf{Q}^-)_{ij}}\right)^{\frac{1}{4}}, \quad (20)$$

which is equivalent to:

$$\big(2(\mathbf{U}\mathbf{U}^T - \mathbf{A})\mathbf{U} + \alpha(\mathbf{U} - \mathbf{F}\mathbf{Q}^+ + \mathbf{F}\mathbf{Q}^-)\big)_{ij}\mathbf{U}_{ij}^4 = 0. \quad (21)$$

It is easy to check that (21) is identical to (19). Both equations require that at least one of the two factors is equal to zero. The first factor in both equations is identical. For the second factor $\mathbf{U}_{ij}$ or $\mathbf{U}_{ij}^4$, if $\mathbf{U}_{ij} = 0$ then $\mathbf{U}_{ij}^4 = 0$, and vice versa. Therefore, (21) and (19) are identical.

Next, we prove the convergence of the update rule in (11) by adopting the auxiliary function approach [12]. We first introduce the definition of auxiliary function as follows.

Definition. *A function $\mathcal{Z}(u, u')$ is an auxiliary function for a given function $\mathcal{J}(u)$ if the conditions $\mathcal{Z}(u, u') \geq \mathcal{J}(u)$ and $\mathcal{Z}(u, u) = \mathcal{J}(u)$ are simultaneously satisfied [12].*

Lemma 1. *If $\mathcal{Z}$ is an auxiliary function for $\mathcal{J}$, $\mathcal{J}$ is non-increasing under the update $u^{(t+1)} = \arg\min_u \mathcal{Z}(u, u^{(t)})$ [12].*

Let $\mathcal{J}(\mathbf{U})$ be the sum of all terms in (9) that contain $\mathbf{U}$, i.e., $\mathcal{J}(\mathbf{U})$ denotes (10). Then we can derive Theorem 3.

Theorem 3. *The auxiliary function of $\mathcal{J}(\mathbf{U})$ is given by:*

$$\begin{aligned}\mathcal{Z}(\mathbf{U}, \mathbf{U}') &= tr(\mathbf{R}^T\mathbf{U}'\mathbf{U}'^T\mathbf{U}') + \frac{\alpha}{2}tr(\mathbf{M}\mathbf{M}^T) \\ &+ \frac{\alpha}{2}tr(\mathbf{U}'\mathbf{U}'^T) + \frac{\alpha}{2}tr(\mathbf{F}\mathbf{Q}^-\mathbf{R}^T) + \frac{3\alpha}{2}tr(\mathbf{F}\mathbf{Q}^-\mathbf{U}'^T) \\ &- 2tr(\mathbf{U}'^T\mathbf{A}\mathbf{Z}) - 2tr(\mathbf{Z}^T\mathbf{A}\mathbf{U}') - 2tr(\mathbf{U}'^T\mathbf{A}\mathbf{U}') \\ &- 2\alpha tr((\mathbf{F}\mathbf{Q}^+)^T\mathbf{Z}) - 2\alpha tr((\mathbf{F}\mathbf{Q}^+)^T\mathbf{U}'),\end{aligned} \quad (22)$$

where $\mathbf{R}_{ij} = \frac{\mathbf{U}_{ij}^4}{\mathbf{U}_{ij}'^3}$, $\mathbf{M}_{ij} = \frac{\mathbf{U}_{ij}^2}{\mathbf{U}_{ij}'}$ *and* $\mathbf{Z}_{ij} = \mathbf{U}_{ij}' \ln \frac{\mathbf{U}_{ij}}{\mathbf{U}_{ij}'}$.

Theorem 3 can be proved using a similar idea to that in [13], [49]. We omit the details due to space limitation. Setting the derivative of $\mathcal{Z}(\mathbf{U}, \mathbf{U}')$ with respect to $\mathbf{U}_{ij}$ to 0, we have:

$$\begin{aligned}\frac{\partial\mathcal{Z}(\mathbf{U}, \mathbf{U}')}{\partial\mathbf{U}_{ij}} =& \frac{\mathbf{U}_{ij}^3}{\mathbf{U}_{ij}'^3}(4\mathbf{U}'\mathbf{U}'^T\mathbf{U}' + 2\alpha\mathbf{U}' + 2\alpha\mathbf{F}\mathbf{Q}^-)_{ij} \\ &- \frac{\mathbf{U}_{ij}'}{\mathbf{U}_{ij}}(4\mathbf{A}\mathbf{U}' + 2\alpha\mathbf{F}\mathbf{Q}^+)_{ij} = 0.\end{aligned} \quad (23)$$

Obviously, the solution of (23) is consistent with (20). Furthermore, according to Lemma 1, we have:

$$\mathcal{J}(\mathbf{U}^{(t+1)}) \leq \mathcal{Z}(\mathbf{U}^{(t+1)}, \mathbf{U}^{(t)}) \leq \mathcal{Z}(\mathbf{U}^{(t)}, \mathbf{U}^{(t)}) = \mathcal{J}(\mathbf{U}^{(t)}).$$

Therefore, we can conclude that the objective function in (9) monotonically decreases under the update rule in (11).

REFERENCES

[1] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *PNAS*, vol. 99, no. 12, pp. 7821–7826, 2002.

[2] W. Cui, Y. Xiao, H. Wang, and W. Wang, "Local search of communities in large graphs," in *SIGMOD*. ACM, 2014, pp. 991–1002.

[3] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *WWW*. ACM, 2010, pp. 631–640.

[4] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.

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

[6] A. Epasto, S. Lattanzi, and R. Paes Leme, "Ego-splitting framework: from non-overlapping to overlapping clusters," in *SIGKDD*. ACM, 2017, pp. 145–154.

[7] J. J. Choong, X. Liu, and T. Murata, "Learning community structure with variational autoencoder," in *ICDM*. IEEE, 2018, pp. 69–78.

[8] Y. Bian, Y. Yan, W. Cheng, W. Wang, D. Luo, and X. Zhang, "On multi-query local community detection," in *ICDM*. IEEE, 2018, pp. 9–18.

[9] 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.

[10] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *PNAS*, vol. 105, no. 4, pp. 1118–1123, 2008.

[11] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 43, 2013.

[12] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2001, pp. 556–562.

[13] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, "Community discovery using nonnegative matrix factorization," *DMKD*, vol. 22, no. 3, pp. 493–521, 2011.

[14] F. Ye, S. Li, Z. Lin, C. Chen, and Z. Zheng, "Adaptive affinity learning for accurate community detection," in *ICDM*. IEEE, 2018, pp. 1374–1379.

[15] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *WSDM*. ACM, 2013, pp. 587–596.

[16] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using bayesian non-negative matrix factorization," *Physical Review E*, vol. 83, no. 6, p. 066114, 2011.

[17] H. Zhang, T. Zhao, I. King, and M. R. Lyu, "Modeling the homophily effect between links and communities for overlapping community detection." in *IJCAI*, 2016, pp. 3938–3944.

[18] X. Shi, H. Lu, and G. Jia, "Adaptive overlapping community detection with bayesian nonnegative matrix factorization," in *DASFAA*. Springer, 2017, pp. 339–353.

[19] Y. Yang, F. Shen, Z. Huang, and H. T. Shen, "A unified framework for discrete spectral clustering." in *IJCAI*, 2016, pp. 2273–2279.

[20] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: Single-view to multi-view," *TIP*, vol. 25, no. 6, pp. 2833–2843, 2016.

[21] Y. Yang, Y. Yang, H. T. Shen, Y. Zhang, X. Du, and X. Zhou, "Discriminative nonnegative spectral clustering with out-of-sample extension," *TKDE*, vol. 25, no. 8, pp. 1760–1771, 2013.

[22] J. Xu, J. Han, and F. Nie, "Multi-view feature learning with discriminative regularization," in *IJCAI*, 2017, pp. 3161–3167.

[23] C.-J. Hsieh and I. S. Dhillon, "Fast coordinate descent methods with variable selection for non-negative matrix factorization," in *SIGKDD*. ACM, 2011, pp. 1064–1072.

[24] M. E. Newman, "Modularity and community structure in networks," *PNAS*, vol. 103, no. 23, pp. 8577–8582, 2006.

[25] R.-H. Li, L. Qin, F. Ye, J. X. Yu, X. Xiao, N. Xiao, and Z. Zheng, "Skyline community search in multi-valued networks," in *SIGMOD*. ACM, 2018, pp. 457–472.

[26] J. Li, X. Wang, K. Deng, X. Yang, T. Sellis, and J. X. Yu, "Most influential community search over large social networks," in *ICDE*. IEEE, 2017, pp. 871–882.

[27] Z. Ding, X. Zhang, D. Sun, and B. Luo, "Low-rank subspace learning based network community detection," *Knowledge-Based Systems*, vol. 155, pp. 71–82, 2018.

[28] Y. Li, K. He, K. Kloster, D. Bindel, and J. Hopcroft, "Local spectral clustering for overlapping community detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 2, p. 17, 2018.

[29] N. Veldt, D. F. Gleich, and A. Wirth, "A correlation clustering framework for community detection," in *WWW*. International World Wide Web Conferences Steering Committee, 2018, pp. 439–448.

[30] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[31] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[32] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, no. 10, p. 103018, 2010.

[33] J. Xie, B. K. Szymanski, and X. Liu, "Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *ICDMW*. IEEE, 2011, pp. 344–349.

[34] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "Demon: a local-first discovery method for overlapping communities," in *SIGKDD*. ACM, 2012, pp. 615–623.

[35] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.

[36] Y. Zhang and D.-Y. Yeung, "Overlapping community detection via bounded nonnegative matrix tri-factorization," in *SIGKDD*. ACM, 2012, pp. 606–614.

[37] H. Zhang, I. King, and M. R. Lyu, "Incorporating implicit link preference into overlapping community detection." in *AAAI*, 2015, pp. 396–402.

[38] F. Ye, C. Chen, and Z. Zheng, "Deep autoencoder-like nonnegative matrix factorization for community detection," in *CIKM*. ACM, 2018, pp. 1393–1402.

[39] T. He, L. Hu, K. C. Chan, and P. Hu, "Learning latent factors for community identification and summarization," *IEEE Access*, vol. 6, pp. 30 137–30 148, 2018.

[40] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang, "Semantic community identification in large attribute networks." in *AAAI*, 2016, pp. 265–271.

[41] B.-H. Shen, S. Ji, and J. Ye, "Mining discrete patterns via binary matrix factorization," in *SIGKDD*. ACM, 2009, pp. 757–766.

[42] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled hybrids of generative and discriminative models," in *CVPR*. IEEE, 2006, pp. 87–94.

[43] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with gaussian kernel," *Neural Computation*, vol. 15, no. 7, pp. 1667–1689, 2003.

[44] H. Zhang, F. Shen, W. Liu, X. He, H. Luan, and T.-S. Chua, "Discrete collaborative filtering," in *SIGIR*. ACM, 2016, pp. 325–334.

[45] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "Uncovering hierarchical and overlapping communities with a local-first approach," *ACM Transactions on Knowledge Discovery from Data*, vol. 9, no. 1, p. 6, 2014.

[46] B.-J. Sun, H. Shen, J. Gao, W. Ouyang, and X. Cheng, "A non-negative symmetric encoder-decoder approach for community detection," in *CIKM*. ACM, 2017, pp. 597–606.

[47] X. Bai, P. Yang, and X. Shi, "An overlapping community detection algorithm based on density peaks," *Neurocomputing*, vol. 226, pp. 7–15, 2017.

[48] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 8, pp. 1706–1712, 2009.

[49] C. H. 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, 2010.