Nonnegative Matrix Factorization Based on Node Centrality for Community Detection

SIXING SU, JIEWEN GUAN, and BILIAN CHEN, Department of Automation, Xiamen University, China and Xiamen Key Laboratory of Big Data Intelligent Analysis and Decision-Making, China
XIN HUANG, Department of Computer Science, Hong Kong Baptist University, China

Community detection is an important topic in network analysis, and recently many community detection methods have been developed on top of the Nonnegative Matrix Factorization (NMF) technique. Most NMF-based community detection methods only utilize the first-order proximity information in the adjacency matrix, which has some limitations. Besides, many NMF-based community detection methods involve sparse regularizations to promote clearer community memberships. However, in most of these regularizations, different nodes are treated equally, which seems unreasonable. To dismiss the above limitations, this article proposes a community detection method based on node centrality under the framework of NMF. Specifically, we design a new similarity measure which considers the proximity of higher-order neighbors to form a more informative graph regularization mechanism, so as to better refine the detected communities. Besides, we introduce the node centrality and *Gini* impurity to measure the importance of nodes and sparseness of the community memberships, respectively. Then, we propose a novel sparse regularization mechanism which forces nodes with higher node centrality to have smaller *Gini* impurity. Extensive experimental results on a variety of real-world networks show the superior performance of the proposed method over thirteen state-of-the-art methods.

CCS Concepts: • **Mathematics of computing** → **Graph algorithms**; • **Computing methodologies** → *Machine learning*; • **Information systems** → **Clustering**

Additional Key Words and Phrases: Community detection, nonnegative matrix factorization, similarity measure, sparse regularization, optimization

ACM Reference format:
Sixing Su, Jiewen Guan, Bilian Chen, and Xin Huang. 2023. Nonnegative Matrix Factorization Based on Node Centrality for Community Detection. *ACM Trans. Knowl. Discov. Data.* 17, 6, Article 84 (February 2023), 21 pages.
https://doi.org/10.1145/3578520

This work was supported in part by Youth Innovation Fund of Xiamen under Grant No. 3502Z20206049, National Natural Science Foundation of China under Grant No. 61836005, and Hong Kong RGC Grant No. 22200320.

Authors' address: S. Su, J. Guan, and B. Chen (corresponding author), Department of Automation, Xiamen University, Xiamen 361005, China, and Xiamen Key Laboratory of Big Data Intelligent Analysis and Decision-Making, Xiamen 361005, China; emails: {23220201151609, jwguan}@stu.xmu.edu.cn, blchen@xmu.edu.cn; X. Huang, Department of Computer Science, Hong Kong Baptist University, Hong Kong, China; email: xinhuang@comp.hkbu.edu.hk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2023 Association for Computing Machinery.

1556-4681/2023/02-ART84 $15.00
https://doi.org/10.1145/3578520

1 INTRODUCTION

Recently, networks have been a prevalent tool to model many real-world entities and their relationships, such as online social networks, collaboration networks, citation networks, protein-protein interaction networks, and so on [14, 35]. One fundamental component of these real networks is their underlying communities. Generally, a community is usually regarded as a group of nodes that are closely connected internally, meanwhile the external links between different communities are sparse. A network may contain several communities. The task of community detection [9, 14, 20] aims to find all these communities, which is also called *graph clustering* [13, 27].

In the literature, there exist a dozens of models to detect communities, due to complex structures and various types of communities over different networks in reality. Traditional graph clustering algorithms [4, 12, 28–30, 33, 36] mostly need to use network information (e.g., betweenness [28]) to identify communities. However, as the scale of networks increases, these traditional algorithms will be too time-consuming to be suitable for efficiently detecting communities. Recently, NMF-based community detection methods are widely used as an effective clustering model [16, 18, 21–23, 25, 31, 32, 34, 38, 41, 43, 44, 46, 51]. The purpose of NMF is to decompose the adjacency matrix of a network into two low-dimensional representations, and one of which can be interpreted as a node-community membership matrix. Based on this matrix, one can infer the underlying community structure in the network. The community detection method based on NMF has the characteristics of high scalability and good interpretability. However, nearly all existing NMF-based community detection algorithms only utilize the adjacency matrix, ignoring the higher-order proximity information of nodes. Besides, in most of these NMF-based community detection algorithms, the weights of sparse regularization (if used) of different nodes are equal, which seems unreasonable. Intuitively, nodes with higher degree shall be paid with more attention.

To tackle the aforementioned limitations, in this article, we first propose a novel node similarity measure and a novel community membership regularization mechanism. Specifically, we design a new node similarity measure to quantify the likeness of topological structures between two nodes, and use it to regularize the detected communities. The proposed similarity measure can utilize more useful information (i.e., higher-order neighbors) than many existing similarity measures. Besides, we design a novel node centrality [8] and *Gini* impurity [2] –based regularization mechanism to promote sparsity of the community memberships of nodes with higher node centrality. Specifically, we adopt the node centrality and *Gini* impurity to measure the importance levels of nodes and confusion levels of community memberships, respectively, and use these two newly introduced measures to regularize the community membership matrix. To reasonably apply the two above-proposed techniques for community detection, we propose a novel community detection algorithm, namely **NMF based on Node Centrality (NCNMF** for short), which fuses them into an NMF framework. Distinct from previous methods, our algorithm combines matrix factorization and graph properties, which can not only take advantage of the high scalability and good interpretability of NMF, but also incorporate the basic characteristics of the graph, has the advantage of being more effective than previous methods. To optimize our proposed NCNMF method, we develop an efficient optimization algorithm. Furthermore, we also conduct theoretical analysis on the convergence of the optimization algorithm and computational complexity of the whole community detection process. Figure 1 shows an overview framework of our proposed NCNMF method.

The main contributions of this article are summarized as follows:

(1) We propose a new similarity measure, namely *h-order weighted Jaccard,* and use it as a vital ingredient to regularize the detected communities. The good side is that our proposed *h*-order weighted Jaccard contains more information of higher-order neighbors than the existing local similarity measures, which is good for community detection.

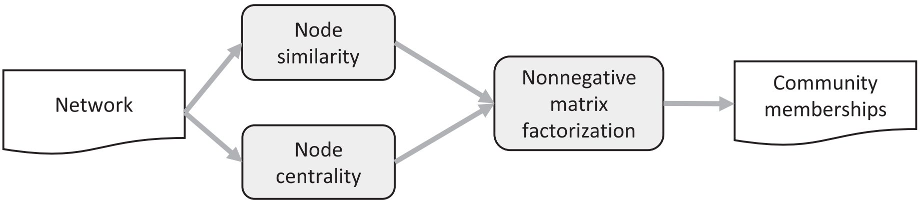


Fig. 1. The framework of the proposed NCNMF method.

(2) We leverage the node centrality and *Gini* impurity to form a community regularization mechanism, which makes the community memberships of nodes with higher node centrality to be sparse. By reducing the importance of nodes with low node centrality and increasing the importance of nodes with high node centrality, the process of non-negative matrix factorization targeted for community detection can be made more effectively.
(3) We propose a novel NCNMF method for community detection, which utilizes the newly proposed similarity measure and sparse regularization mechanism under the NMF framework.
(4) We derive an efficient optimization algorithm based on the gradient descent method with theoretically guaranteed convergence to solve NCNMF. Moreover, the computational complexity of the whole community detection process is analyzed. It scales cubically with the number of nodes in the network, which is the same as many existing NMF-based algorithms and thus guarantees its efficiency.
(5) We conduct extensive experiments on eight real-world benchmark networks to test NCNMF, in comparison with thirteen state-of-the-art community detection methods. Experimental results not only show the great superiority of NCNMF, but also demonstrate that NCNMF strikes a good balance between effectiveness and efficiency. Moreover, we validate the theoretical analysis and conduct the sensitivity analysis.

The rest of this article is organized as follows: In Section 2, we give a brief review of related work. Next, we present the preliminaries of community detection in Section 3. The proposed NCNMF model is then detailed in Section 4, followed by the optimization algorithm and theoretical analysis in Section 5. In Section 6, extensive experimental results are reported. This article is finally concluded in Section 7.

2 RELATED WORK

In this section, we provide a brief review about related traditional graph clustering algorithms and NMF-based community detection methods.

2.1 Traditional Graph Clustering Algorithms

Traditional graph clustering algorithms utilize graph characteristics to obtain tight clustering of nodes. Among them, the pioneers, Newman et al. [28], first proposed the most widely adopted method for community detection. Specifically, Newman et al. discovered the community structure through one of a number of possible "betweenness" measures, and proposed a measure called *modularity* for measuring the strength of the detected community structure. Although modularity is a well-designed metric, it still has room for efficiency. In order to speed up modularity-based community detection, Blondel et al. [4] proposed the Louvain method to extract the community structure of large networks, with the help of a heuristic method to speed up modularity optimization. In order to exploit the overlapping nature of communities, Palla et al. [29] proposed the first algorithm that can find overlapping communities in networks, meaning that nodes can belong to multiple communities at the same time. Subsequently, many heuristic community detection methods are developed.

For example, Raghavan et al. [33] investigated a simple **label propagation algorithm (LPA)** that only uses the network structure; Perozzi et al. [30] proposed a novel approach for learning latent representations of vertices called *DeepWalk*, which uses random walks to approximate the pointwise mutual information matrix obtained by pooling normalized adjacency matrix powers; Shao et al. [36] proposed a new community detection algorithm called *Attractor*, which automatically spots communities in a network by examining the changes of "distances" among nodes; Epasto et al. [12] proposed the Ego-Splitting framework to detect communities in complex networks. Berahmand et al. [3] proposed a local approach based on the detection and expansion of core nodes, which has the ability to detect all communities in a network using local information as well as identify various roles of nodes. Besides, evolutionary algorithm-based community detection methods have also attracted much attention recently. For example, Lyu et al. [24] proposed an evolutionary-based local community detection algorithm, which uses the entire acquired information to detect local communities in complex networks; Teng et al. [39] proposed a multi-objective evolutionary algorithm based on similar attributes for the detection of overlapping communities in attributed networks. Furthermore, aiming to exploit the higher-order connections of nodes, Huang et al. [15] proposed a method of higher-order connection enhanced multi-view modularity, which is to enhance the intra-community connection of each view by using the higher-order connectivity structure.

2.2 Learning Model based Community Detection

Learning model based community detection algorithms try to learn compact node representations to determine the underlying community structure. NMF-based community detection methods attempt to learn community structures by factorizing the adjacency matrix and have good interpretability [19] and many applications. As pioneers, Psorakis et al. [32] presented a probabilistic approach for community detection, which utilizes a Bayesian nonnegative matrix factorization model to extract community information from a network for overlapping community detection. Cai et al. [6] developed a **graph based approach (GNMF)** for parts-based data representation, which constructs an affinity graph to encode the geometrical information and seeks a matrix factorization to preserve the graph structure. Subsequently, many matrix tri-factorization-based community detection methods have been proposed. For example, Zhang et al. [51] proposed a method called *bounded nonnegative matrix tri-factorization*, which can explicitly model and learn the community memberships of nodes as well as the interactions among communities; Jin et al. [16] proposed the graph regularized nonnegative matrix tri-factorization model, which utilizes the spectral properties of the network to detect communities. To fully preserve the symmetric nature of the adjacency matrix, Kuang et al. [18] developed the **symmetric NMF (SymmNMF)** model, which decomposes the second power of the normalized adjacency matrix based on symmetric nonnegative matrix factorization. Besides, Filippo et al. [31] argued that strict orthogonality of the community membership matrix is important and proposed the **orthogonal nonnegative matrix factorization (ONMF)** model, which imposes both nonnegativity and orthogonality constraints on the community membership matrix. Wang et al. [41] argued that the structure and inherent properties of the network should be preserved in the network embedding, and proposed a **modularized nonnegative matrix factorization (MNMF)** model to incorporate the community structure into network embedding. By recognizing that NMF is just a decoder, Sun et al. [38] proposed a **nonnegative symmetric encoder-decoder approach (NNSED)** for community detection, and explicitly integrated a decoder and an encoder into a unified loss function. Ye et al. [45] proposed a novel **deep autoencoder-like NMF model (DANMF)** for community detection, which had extended Sun's NNSED to a deep autoencoder-like architecture, this architecture empowers DANMF to learn the hierarchical mappings between the original network and the final community assignment with implicit low- to high-level hidden attributes of the original network learnt in the intermediate layers. To encode inherent community structures into node embeddings, Li et al. [22] proposed the CDE model based on the observation of densely connected structures in communities. For detecting communities in a dynamic setting, Ma et al. [25] proposed two evolutionary nonnegative matrix factorization frameworks to detect dynamic communities. For detecting communities in a hypergraph, Li et al. [21] proposed an **edge enhancement approach for motif-aware community detection (EdMot)**, which creates a graph composed of higher-order motifs, and clusters the created graph by the Louvain method. For detecting communities through node homophily (i.e., node similarity), Ye et al. [44] proposed the **homophily preserving NMF (HPNMF)** model, which can reflect the inherent properties of communities. Besides, Ye et al. [47] intended to adaptively learn an affinity matrix, which can capture the intrinsic similarity between nodes accurately, so as to benefit the community detection results. Moreover, to better capture the overlapping nature of communities, Ye et al. [46] proposed the discrete nonnegative matrix factorization model to seek for a discrete (binary) community membership matrix directly. Recently, Rozemberczki et al. [34] proposed a graph-embedding algorithm called *GEMSEC*, which learns community structures and node embeddings simultaneously. As another line of research, semi-supervised community detection methods have also attracted remarkable attention recently. For example, Liu et al. [23] presented a semi-supervised nonnegative matrix factorization model for community detection, which combines the idea of graph regularization with the pairwise constraints; Wu et al. [43] proposed a novel SymmNMF-based semi-supervised clustering method, namely *pairwise constraint propagation-induced SymmNMF*, which can learn the similarity and assignment matrices adaptively and simultaneously. However, most of the above NMF-based community detection methods only work on the original network topology (i.e., the adjacency matrix), and regularize different nodes equally if sparse regularization is involved, leading to poor results due to neglecting the higher-order proximity of the graph and individual discrepancy of nodes. Unlike these methods, our method utilizes the newly proposed similarity measure and sparse regularization mechanism, which can further improve the quality of community detection.

Recently, with the rise of deep learning technology, many community detection models based on neural network have been proposed. Zhang et al. [50] proposed an **adaptive graph convolution method (AGC)** for attributed graph clustering that exploits node relations and the diversity of graphs to capture global cluster structure. Zhao et al. [52] proposed an inductive embedding model, which utilized a multi-core convolutional neural network and a semi-supervised learning mechanism to learn the robust representations for an attributed network. De et al. [10] proposed an innovative approach for semi-supervised community detection, which exploited convolutional neural networks and different properties of a network to build the network connections over particular sparse matrices. Since most of these models are designed for attribute networks, and the emphasis of our research is on community detection in attribute-free networks, we only briefly review them and refer interested readers to a recent article [37] including deep learning models based on deep neural networks, deep nonnegative matrix factorization and deep sparse filtering.

3 PRELIMINARIES

In this section, we first introduce the main notations and terminologies used in this article. Then, we present several existing similarity measures, the symmetric NMF model, and the homophily-preserving NMF model.

3.1 Notations

We use boldface uppercase letters to represent matrices, boldface lowercase letters to represent vectors, and italic lowercase letters to represent scalar values. An element of a vector $\mathbf{x}$ is represented by x_i, and an element of a matrix $\mathbf{X}$ is represented interchangeably by x_{ij} and $(\mathbf{X})_{ij}$. We use

Table 1. Summary of Notations

Symbol	Description
$\mathbf{X} \in \mathbb{R}^{m\times n}$	a matrix
$\mathbf{x}$	a vector
$\mathrm{diag}(\mathbf{x})$	a diagonal matrix whose diagonal entries are composed of $\mathbf{x}$
$\|\|\mathbf{x}\|\|_2$	Euclidean norm of the vector $\mathbf{x}$
$\mathcal{G}$	an undirected and unweighted network
$\mathcal{V}$	the nodes in $\mathcal{G}$
$\mathcal{E}$	the edges in $\mathcal{G}$
n	the number of the nodes in $\mathcal{G}$
m	the number of the edges in $\mathcal{G}$
$\mathbf{A} \in \{0,1\}^{n\times n}$	Boolean adjacency matrix
a_{ij}	the (i,j)th entry of matrix $\mathbf{A}$
$\Gamma(v_i)$	the neighbor set of node v_i and itself
C	the detected disjoint communities
c_i	the ith community, where $1 \le i \le k$
$\xi(v_i)$	the index of the community containing v_i
x_{ij} and $(\mathbf{X})_{ij}$	an element of a matrix $\mathbf{X}$
x_i	an element of a vector $\mathbf{x}$
$\mathbf{1}_d$	the d-dimensional all-one vector
$\mathbf{I}_d \in \mathbb{R}^{d\times d}$	the identity matrix
$\mathbf{E}_{m\times n} \in \mathbb{R}^{m\times n}$	the all-one matrix
$\mathbf{x}_{i:}$	the ith row vector of $\mathbf{X}$
$\mathbf{x}_{:j}$	the jth column vector of $\mathbf{X}$
$\mathrm{Tr}(\mathbf{X})$	the trace of $\mathbf{X}$ if it is square
$\mathbf{X}^T$	transpose of $\mathbf{X}$
$\|\|\mathbf{X}\|\|_F$	the Frobenius norm of $\mathbf{X}$
S	the logical statement
$\mathbb{1}(S)$	the indicator function of the logical statement S

$\mathbf{1}_d$ to denote a d-dimensional all-one vector, $\mathbf{I}_d$ to denote an identity matrix in $\mathbb{R}^{d\times d}$, and $\mathbf{E}_{m\times n}$ to denote an all-one matrix in $\mathbb{R}^{m\times n}$. We use $\mathrm{diag}(\mathbf{x})$ to represent a diagonal matrix whose diagonal entries are composed of $\mathbf{x}$. For a vector $\mathbf{x} \in \mathbb{R}^n$, we use $||\mathbf{x}||_2$ to represent its Euclidean norm. For a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, we use $\mathbf{x}_{i:}$ and $\mathbf{x}_{:j}$ to represent its ith row vector and jth column vector, respectively. Besides, we adopt $\mathrm{Tr}(\mathbf{X})$ to denote the trace of $\mathbf{X}$ if it is square (i.e., $m = n$), $\mathbf{X}^T$ to denote the transpose of $\mathbf{X}$, and $||\mathbf{X}||_F$ to denote the Frobenius norm of $\mathbf{X}$. We use $\mathbb{1}(S)$ to denote the indicator function of the logical statement S, i.e., $\mathbb{1}(S) = 1$ if S is true, or $\mathbb{1}(S) = 0$ otherwise.

Next, we introduce some notations used in networks. We consider an undirected and unweighted network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ represents all the nodes in $\mathcal{G}$ and the edge set $\mathcal{E} = \{e_1, e_2, \ldots, e_m\} \subseteq \mathcal{V} \times \mathcal{V}$ represents all the edges in $\mathcal{G}$, respectively. We denote by $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$. The network $\mathcal{G}$ can be represented by a Boolean adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$, whose (i,j)th entry $a_{ij} = \mathbb{1}((v_i, v_j) \in \mathcal{E})$. The symbol $\Gamma(v_i)$ is defined as the neighbor set of node v_i and itself, i.e., $\Gamma(v_i) = \{v_j : (v_i, v_j) \in \mathcal{E}\} \cup \{v_i\}$. Besides, assume that we know *a priori* that there are k communities to be detected in $\mathcal{G}$, then our community detection algorithm can return k detected disjoint communities as $C = \{c_i \mid c_i \neq \emptyset, \bigcup_{i=1}^{k} c_i = \mathcal{V}, \text{and } c_i \cap c_j = \emptyset, \forall\, i \neq j\}$, where c_i denotes the ith community for $1 \le i \le k$. We use $\xi(v_i)$ to denote the index of the community containing v_i, i.e., $v_i \in c_{\xi(v_i)}$. For the convenience of reading, the notations of this article are summarized in Table 1.

3.2 Similarity Measures

We review different structural similarity metrics to measure the similarity of nodes. The similarity of a pair of nodes (e.g., v_i and v_j) is a measure about whether the two nodes should be allocated into the same community or not. The existing similarity measures are categorized into two types of local similarity measures and global similarity measures. The details are discussed below.

Local Similarity Measures. Local similarity measures include the naive similarity measure [44], the Common Neighbor measure [27], the Cosine measure [49], and the Jaccard measure [49], and so on. The Common Neighbor measure is defined as the number of common neighbors between two nodes. The Cosine measure is a measure of similarity between two sequences of numbers (in our case, the adjacency lists), which is defined as the dot product of the vectors divided by the product of their lengths. It follows that the Cosine measure does not depend on the magnitudes of the vectors, but only on their angle. Besides, the Cosine measure is basically used in calculating document similarity. Since the Jaccard measure is an enhanced version of the Common Neighbor measure, researchers now basically use the Jaccard measure instead of the Common Neighbor measure to calculate nodes similarity. Next, we only focus on the naive similarity measure and the Jaccard similarity measure.

(1) The naive similarity measure is defined as

$$Sim_{\text{naive}}(v_i, v_j) = \mathbb{1}((v_i, v_j) \in \mathcal{E}).$$

(2) The Jaccard similarity measure is defined as

$$Sim_{\text{Jaccard}}(v_i, v_j) = \frac{|\Gamma(v_i) \cap \Gamma(v_j)|}{|\Gamma(v_i) \cup \Gamma(v_j)|}. \tag{1}$$

The naive similarity $Sim_{\text{naive}}(v_i, v_j)$ is the simplest similarity measure based on the edge connections. The naive similarity matrix is in fact equivalent to the adjacency matrix, which provides great convenience in subsequent calculations. The Jaccard similarity $Sim_{\text{Jaccard}}(v_i, v_j)$ utilizes the neighbor sets of nodes to calculate the similarity. It is defined as the size of the intersection divided by the size of the union of two nodes' neighbor sets $\Gamma(v_i)$ and $\Gamma(v_j)$.

Local similarity measures have two main advantages. One is that they provide an intuitive way to characterize the node similarity. The other one is that they are convenient to calculate and consume less time. However, the shortcomings of local similarity measures are also obvious. They only consider the impact of directly connected neighbors on the similarity computations, ignoring a lot of useful information, such as the higher-order neighbors of nodes [42].

Global Similarity Measures. Global similarity measures include the famous Katz index [17], LHN-II index [1], and the like. The Katz index is computed by searching the graph for paths and adding the counts of each path length weighted by user specified weights, which is a graph-based computational method and computes similarities between nodes in a global network. The LHN-II index is proposed based on the regular equivalence [1], which indicates nodes x and y are similar when the neighbors of node x are similar to node y, that is, the similarity of the nodes is transitive. In addition, the similarity matrix constructed in the subspace clustering model [40] also belongs to the global similarity measure. The most common method for building similarity matrices in these models is the fully connected method, where all points have a weight value greater than 0. Different kernel functions can be selected to define the edge weights, the most commonly used one is the Gaussian kernel function. They can overcome the shortcomings of local similarity measures by taking the whole network topology into consideration, but their time complexities are higher than those of local similarity measures, which are not scalable on large networks [42].

As the scale of the networks becomes larger, researchers prefer to choose local similarity measures to detect communities, since they consume less time.

3.3 Symmetric NMF

We introduce the **symmetric NMF model (SymmNMF)** [18]. SymmNMF tries to reconstruct the adjacency matrix $\mathbf{A}$ by two identical factor matrices, representing node-community memberships, and it serves as a building block of our proposed NCNMF method. Specifically, we denote the community membership matrix as $\mathbf{H} \in \mathbb{R}^{n \times k}$, where each element h_{il} reflects the tendency that $v_i \in c_l$, SymmNMF models the expected number of edges between v_i and v_j as $\hat{a}_{ij} = \sum_{l=1}^{k} h_{il} h_{jl} = \mathbf{h}_{i:} \mathbf{h}_{j:}^T$. It is obvious that, for any $(v_i, v_j) \in \mathcal{E}$, $\hat{a}_{ij}$ should be consistent with a_{ij} (so that the edge modeling is precise), which gives rise to the following optimization problem of SymmNMF:

$$\min_{\mathbf{H} \geq 0} \ \|\mathbf{A} - \mathbf{H}\mathbf{H}^T\|_F^2.$$

3.4 Homophily Preserving NMF

We introduce the **homophily preserving NMF model (HPNMF)** [44]. The basic idea of HPNMF is to take node similarity into consideration. The more similar two nodes are, the more similar their community memberships should be. According to [44], given a well-defined similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, a good way to implement the above idea is to optimize the following optimization problem:

$$\min_{\mathbf{H} \geq 0, \mathbf{H}^T\mathbf{H} = \mathbf{I}_k} \ \|\mathbf{A} - \mathbf{H}\mathbf{H}^T\|_F^2 + \lambda \operatorname{Tr}(\mathbf{H}^T \mathbf{L} \mathbf{H}) + \gamma \, \|\mathbf{H}\mathbf{1}_k\|_2^2 \, ,$$

where λ is a positive parameter controlling the importance of the node similarity information, γ is also a positive parameter controlling the sparse regularization on $\mathbf{H}$, the regularizer $\|\mathbf{H}\mathbf{1}_k\|_2^2 = \sum_{i=1}^{n} (\mathbf{h}_{i:} \mathbf{1}_k)^2 = \sum_{i=1}^{n} \|\mathbf{h}_{i:}\|_1^2$ is actually imposing ℓ_1-norm regularization on rows of $\mathbf{H}$ to make them sparser, and $\mathbf{L}$ is the graph Laplacian matrix defined as

$$\mathbf{L} = \mathbf{S}' - \mathbf{S}, \tag{2}$$

where $\mathbf{S}$ is the above-mentioned similarity matrix, $\mathbf{S}'$ is a diagonal matrix with $s'_{ii} = \sum_{j=1}^{n} s_{ij}$.

4 NONNEGATIVE MATRIX FACTORIZATION BASED ON NODE CENTRALITY

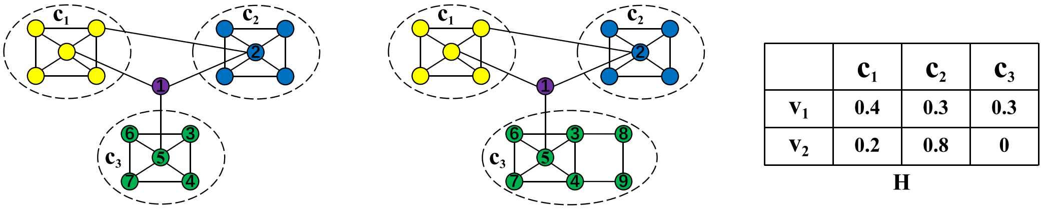
In this section, we first propose a new similarity measure, namely h-order weighted Jaccard, which utilizes the structural information of h-order neighborhood in a network. Then, we develop a novel sparse regularization mechanism based on the node centrality and *Gini* impurity. Combining these two techniques, we propose our NCNMF model for the community detection problem.

4.1 The Formula of h-order Weighted Jaccard Similarity

Let h be a nonnegative constant representing the number of hops and p be an integer number in the range of $[1, h]$. For a given node v_i, we denote its directly-connected neighbors as $\Gamma(v_i)$, and its higher-order neighbors as $\Gamma_p(v_i) = \{v_j \in \mathcal{V} : v_j$ locates in the p-hop neighborhood of v_i and $2 \leq p \leq h\}$. Based on these two neighbor sets, we propose a new measure for calculating node similarity, namely h-order weighted Jaccard (denoted as w-Jaccard) similarity. Its specific calculation is as follows

$$Sim_{\text{w-Jaccard}}(v_i, v_j | h) = \sum_{p=1}^{h} \sum_{q=1}^{h} \Omega_{p,q} \frac{|\Gamma_p(v_i) \cap \Gamma_q(v_j)|}{|\Gamma_p(v_i) \cup \Gamma_q(v_j)|}, \tag{3}$$

where $\Omega_{p,q}$ is the weight of the sub-similarity $|\Gamma_p(v_i) \cap \Gamma_q(v_j)| / |\Gamma_p(v_i) \cup \Gamma_q(v_j)|$. In our scheme, $\Omega_{p,q}$ decreases as either p or q increases. The parameter h here is suggested to be set as a constant in $\{1, 2, 3\}$ [42], indicating that node v_i can take h-order neighbors at most.



	c1	c2	c3
v1	0.4	0.3	0.3
v2	0.2	0.8	0

(a) The original network $\mathcal{G}$.

(b) The new network $\hat{\mathcal{G}}$ which adds two nodes $\{v_8, v_9\}$ and three edges $\{(v_3, v_8), (v_4, v_9), (v_8, v_9)\}$ on $\mathcal{G}$.

(c) Community memberships.

Fig. 2. An example of h-order weighted Jaccard similarity and *Gini* impurity.

As mentioned in Section 3.2, the normal Jaccard similarity in Equation (1) only utilizes the information of directly connected neighbors. This neglects some useful information, such as the higher-order neighbors of nodes, resulting in a limited accuracy of node similarity. We use the following example to concretely illustrate the improvement of the w-Jaccard similarity over the Jaccard similarity.

Example 1. Consider Figure 2(a) and Figure 2(b), the supergraph $\hat{\mathcal{G}}$ adds two new nodes $\{v_8, v_9\}$ and three new edges $\{(v_3, v_8), (v_4, v_9), (v_8, v_9)\}$ on top of $\mathcal{G}$. Let us focus on the community c_3 in both networks. Intuitively, the relationship between nodes v_3 and v_4 in $\hat{\mathcal{G}}$ in Figure 2(b) should be closer than that in $\mathcal{G}$ in Figure 2(a). When we apply the w-Jaccard similarity (i.e., Equation (3)) with $h = 2$ and $\Omega_{p,q} = (5/8)^{pq}$ [42] on the two nodes, we get $Sim_{\text{w-Jaccard}}(v_3, v_4) = 73/128$ in Figure 2(a) and $Sim_{w\text{-Jaccard}}(v_3, v_4) = 361/605$ in Figure 2(b), which is in line with reality. However, if we apply the normal Jaccard similarity (i.e., Equation (1)) on them, we find $Sim_{\text{Jaccard}}(v_3, v_4) = 3/5$ in Figure 2(a) and $Sim_{\text{Jaccard}}(v_3, v_4) = 3/7$ in Figure 2(b), which contradicts the reality.

ALGORITHM 1: Calculation of the w-Jaccard Similarity

```
   Input: Network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and parameter $h$.
   Output: The $w$-Jaccard similarity matrix S.
1  Initialize $S \leftarrow \{0\}^{n \times n}$, $\Gamma_p(v_i) \leftarrow \varnothing$ for all $p \in \{1, \dots, h\}$ and $v_i \in \mathcal{V}$;
2  for all $(v_i, v_j) \in \mathcal{E}$ do
3      $\Gamma_1(v_i) \leftarrow \Gamma_1(v_i) \cup \{v_j\}$;
4  for all $v_i \in \mathcal{V}$ do
5      for $p \in \{2, \dots, h\}$ do
6          for all $(v_k, v_j) \in \mathcal{V} \times \Gamma_{p-1}(v_i)$ do
7              $\Gamma_p(v_i) \leftarrow \Gamma_p(v_i) \cup \{v_k\}$ if $(v_k, v_j) \in \mathcal{E}$;
8  for all $(v_i, v_j) \in \mathcal{V} \times \mathcal{V}$ do
9      for $(p, q) \in \{1, \dots, h\} \times \{1, \dots, h\}$ do
10         $s_{ij} \leftarrow s_{ij} + \Omega_{p,q} \frac{|\Gamma_p(v_i) \cap \Gamma_q(v_j)|}{|\Gamma_p(v_i) \cup \Gamma_q(v_j)|}$;
11 return S.
```

Algorithm 1 summarizes the whole procedure of constructing the similarity matrix based on w-Jaccard. Given a network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and parameter h, Algorithm 1 first initializes the similarity matrix S and the higher-order neighbors $\Gamma_p(v_i)$ for all $v_i \in \mathcal{V}$ (line 1), and then gets $\Gamma_1(v_i)$ for all

ALGORITHM 2: Calculation of the Degree Matrix

```
  Input: Network $\mathcal{G}(\mathcal{V},\mathcal{E})$.
  Output: The degree matrix $\mathbf{D}$.
1 Initialize $\mathbf{D} \leftarrow \{0\}^{n\times n}$, $d(v_i) \leftarrow 0$ for all $v_i \in \mathcal{V}$;
2 for all $(v_i, v_j) \in \mathcal{E}$ do
3     $d(v_i) \leftarrow d(v_i) + 1$;
4     $d(v_j) \leftarrow d(v_j) + 1$;
5 for all $v_i \in \mathcal{V}$ do
6     $d_{ii} \leftarrow d(v_i)$;
7 return $\mathbf{D}$.
```

$v_i \in \mathcal{V}$ (lines 2-3). Afterwards, it calculates $\Gamma_p(v_i)$ for all $v_i \in \mathcal{V}$ when $p = 2, \dots, h$ (lines 4-7). Eventually, it extracts the w-Jaccard similarity matrix $\mathbf{S}$ via Equation (3) (lines 8-10) and returns the matrix $\mathbf{S}$ (line 11).

4.2 Sparse Regularization Based on Node Centrality

In social networks, core nodes of a community usually have higher degrees and larger node centrality. Node centrality measures the central property of a node, which reflects the importance of a node in the network. Usually, the larger the centrality of a node v_i, the more likely it will form a community with its neighbor nodes. Inspired by the phenomenon, we in this part propose a new sparse regularization scheme based on node centrality, i.e., enforcing the community memberships of nodes with higher node centrality to be sparser.

In an undirected graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, we utilize the degree of a node to represent its centrality [8], which is defined as

$$NC(v_i) = d(v_i),$$

where $d(v_i)$ is the degree of node v_i. The degree matrix $\mathbf{D} = \mathrm{diag}([d(v_1), d(v_2), \dots, d(v_n)])$ is used to represent the centrality of each node. Algorithm 2 summarizes the whole procedure of constructing the degree matrix. Given a network $\mathcal{G}(\mathcal{V},\mathcal{E})$, Algorithm 2 first initializes the degree matrix $\mathbf{D}$ and the nodes' degree counters $d(v_i)$ for all $v_i \in \mathcal{V}$ (line 1), and then loops over $(v_i, v_j) \in \mathcal{E}$ to update $d(v_i)$'s until they are exactly node degrees (lines 2-4). Eventually, it assigns $d(v_i)$'s to the degree matrix $\mathbf{D}$ (lines 5-6) and returns the matrix $\mathbf{D}$ (line 7).

Besides, the *Gini* impurity [2] is defined as

$$Gini(v_i) = \sum_{l=1}^{k} p_{il}(1 - p_{il}),$$

where p_{il} is the probability that node v_i belongs to the lth community and k is the number of communities. It is introduced to judge whether the community membership of a node is sparse, i.e., a lower value of the *Gini* impurity implies a sparser community membership. In the contexts of NMF-based community detection, we propose the following formula to evaluate whether the community membership of v_i is chaotic.

$$Gini(v_i) = \sum_{j=1}^{k} h_{ij}(1 - h_{ij}) = [\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)]_{ii},$$

where $\mathbf{H} = (h_{ij}) \in \mathbb{R}_+^{n\times k}$ is the community membership matrix with $\mathbf{H}\mathbf{1}_k = \mathbf{1}_n$.

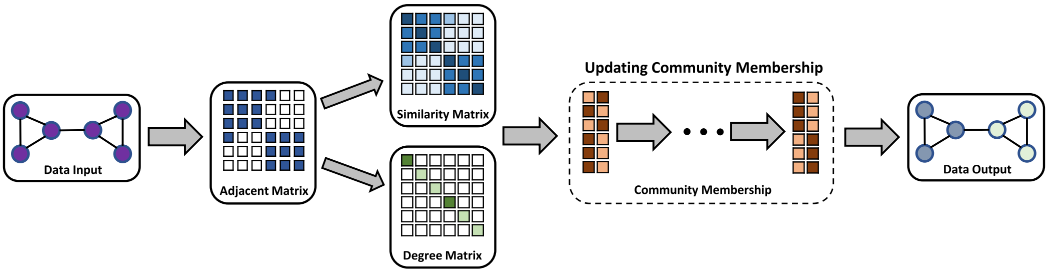


Fig. 3. The flowchart of the proposed NCNMF method.

Example 2. Consider again the network $\mathcal{G}$ in Figure 2(a) and the corresponding community memberships in Figure 2(c). In accordance with the above formula, the community membership of node v_1 is $[0.4, 0.3, 0.3]$, therefore $Gini(v_1) = 0.4 \times (1 - 0.4) + 0.3 \times (1 - 0.3) + 0.3 \times (1 - 0.3) = 0.66$. Similarly, the community membership of node v_2 is $[0.2, 0.8, 0]$, therefore $Gini(v_2) = 0.2 \times (1 - 0.2) + 0.8 \times (1 - 0.8) + 0 \times (1 - 0) = 0.32$. The results show that the community membership of node v_2 is purer than that of node v_1. It is in accord with the reality: it is difficult to distinguish which community that node v_1 belongs to, but node v_2 can be easily identified as belonging to c_2.

The focus of our sparse regularization scheme is to reduce the value of $Gini(v_i)$ for those v_i with larger node centrality, thereby making their community memberships sparser and increasing their separability. Specifically, the sparse regularization term involved in our model is formulated as

$$\mathrm{Tr}[\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)\mathbf{D}].$$

4.3 Nonnegative Matrix Factorization Based on Node Centrality

Based on the above discussions, a variant NMF model based on node centrality, i.e., NCNMF, is proposed to detect disjoint communities. Figure 3 shows a flowchart of our proposed NCNMF method on how to detect communities. Initially, an adjacency matrix is constructed according to the input data, and then the similarity matrix and degree matrix are obtained from the adjacency matrix. Subsequently, we utilize the NCNMF model to obtain the community membership matrix and thus divide the community. The corresponding optimization model is formulated as

$$\min_{\substack{\mathbf{H}\geq 0,\\ \mathbf{H}\mathbf{1}_k=\mathbf{1}_n}} J = \|\mathbf{A} - \mathbf{H}\mathbf{H}^T\|_F^2 + \lambda\,\mathrm{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}) + \alpha\,\mathrm{Tr}[\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)\mathbf{D}]. \tag{4}$$

The explanations of each term and constraint are as follows:

(1) $\|\mathbf{A} - \mathbf{H}\mathbf{H}^T\|_F^2$ **:** $\|\mathbf{A} - \mathbf{H}\mathbf{H}^T\|_F^2$ is the objective function of the SymmNMF, which serves as the backbone of our proposed community detection method NCNMF.
(2) $\lambda\,\mathrm{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H})$ **:** $\mathrm{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H})$ aims to make two similar nodes be assigned to the same community. Here, λ is a positive parameter controlling the importance of the node similarity information, $\mathbf{L}$ is the graph Laplacian matrix defined by Equation (2) (in which $\mathbf{S}$ is replaced by our w-Jaccard similarity matrix).
(3) $\alpha\,\mathrm{Tr}[\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)\mathbf{D}]$ **:** Recall that $[\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)]_{ii}$ represents the *Gini* impurity of the community membership of node v_i, thus the term $\mathrm{Tr}[\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)\mathbf{D}]$ aims to decrease $Gini(v_i)$ for all v_i with higher node centrality. In this way, nodes with higher node centrality will be forced to possess purer (i.e., sparser) community memberships. This term plays the role of sparse regularization. Here, α is a positive parameter controlling the importance of this regularization.

ALGORITHM 3: Community Detection Based on NCNMF

```
Input: Network $\mathcal{G}(\mathcal{V}, \mathcal{E})$, parameters $\lambda$, $\alpha$, $\delta$, $\beta$, $h$, the maximum number of iterations $T$, the
       number of communities to be detected $k$.
Output: The $k$ detected communities in $\mathcal{G}$.
Extract adjacency matrix $\mathbf{A}$ from $\mathcal{G}$ and compute degree matrix $\mathbf{D}$ via Algorithm 2;
Get similarity matrix $\mathbf{S}$ via Algorithm 1 with input adjacency matrix $\mathbf{A}$ and parameter $h$;
Set $t \leftarrow 0$, randomly initialize $\mathbf{H}_t \geq 0$, and compute the initial value of the objective function $J_t$;
while $t < T$ do
    Update $\mathbf{H}_t$ according to the updating rule (7);
    $t \leftarrow t + 1$;
    Update the value of the objective function $J_t$;
    if stopping criterion is satisfied then
        break;
for all $v_i \in \mathcal{V}$ do
    Set $\xi(v_i) \leftarrow \arg\max_j (h_t)_{ij}$;
    Update $c_{\xi(v_i)} \leftarrow c_{\xi(v_i)} \cup \{v_i\}$;
return the $k$ detected communities $C = \{c_1, c_2, \ldots, c_k\}$.
```

(4) $\mathbf{H1}_k = \mathbf{1}_n$ **:** This constraint restricts the sum of each row of the community membership matrix $\mathbf{H}$ to be 1, which is the premise of the *Gini* impurity.

However, as the objective function of Problem (4) is a quartic function of $\mathbf{H}$, and there are two constraints simultaneously imposed on $\mathbf{H}$, it is actually hard to get the analytical solution of this optimization problem.

5 OPTIMIZATION AND THEORETICAL ANALYSIS

In this section, we derive an optimization algorithm based on the gradient descent method to solve Problem (4). Then, we analyze the convergence of this algorithm and computational complexity of the whole NCNMF-based community detection process.

5.1 Solution Method

Instead of optimizing Problem (4) directly, we equivalently reformulate it as

$$\min_{\mathbf{H} \geq 0} \ J = \|\mathbf{A} - \mathbf{H}\mathbf{H}^T\|_F^2 + \lambda \operatorname{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}) + \alpha \operatorname{Tr}[\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)\mathbf{D}] + \delta \ \|\mathbf{H}\mathbf{1}_k - \mathbf{1}_n\|_2^2 \, , \tag{5}$$

where δ is a large positive constant to ensure that the sum of each row of the community membership matrix $\mathbf{H}$ equals to 1. To optimize Problem (5), we first rewrite it in the trace form as

$$\begin{aligned}\min_{\mathbf{H} \geq 0} \ J = &\operatorname{Tr}(\mathbf{A}\mathbf{A}^T - 2\mathbf{A}\mathbf{H}\mathbf{H}^T + \mathbf{H}\mathbf{H}^T\mathbf{H}\mathbf{H}^T) + \lambda \operatorname{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}) \\ &+ \alpha \operatorname{Tr}[\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)\mathbf{D}] + \delta \operatorname{Tr}(\mathbf{H}\mathbf{E}_{k\times k}\mathbf{H}^T - 2\mathbf{H}\mathbf{E}_{k\times n} + \mathbf{E}_{n\times n}).\end{aligned}$$

Then, we introduce a Lagrange multiplier matrix $\Phi \in \mathbb{R}_+^{n\times k}$ for the nonnegative constraints on $\mathbf{H}$, which leads to the following Lagrangian function

$$\begin{aligned}J_\Phi = &\operatorname{Tr}(\mathbf{A}\mathbf{A}^T - 2\mathbf{A}\mathbf{H}\mathbf{H}^T + \mathbf{H}\mathbf{H}^T\mathbf{H}\mathbf{H}^T) + \lambda \operatorname{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}) \\ &+ \alpha \operatorname{Tr}[\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)\mathbf{D}] + \delta \operatorname{Tr}(\mathbf{H}\mathbf{E}_{k\times k}\mathbf{H}^T - 2\mathbf{H}\mathbf{E}_{k\times n} + \mathbf{E}_{n\times n}) - \operatorname{Tr}(\Phi\mathbf{H}^T).\end{aligned}$$

The partial derivative of J_Φ with respect to $\mathbf{H}$ is

$$\frac{\partial J_\Phi}{\partial \mathbf{H}} = -\,4\mathbf{AH} + 4\mathbf{HH}^T\mathbf{H} + 2\lambda\mathbf{LH} + \alpha\mathbf{DE}_{k\times n}^T - 2\alpha\mathbf{DH} + 2\delta\mathbf{HE}_{k\times k} - 2\delta\mathbf{E}_{k\times n}^T - \Phi.$$

By setting $\partial J_\Phi/\partial\mathbf{H} = 0$, we obtain

$$\Phi = -\,4\mathbf{AH} + 4\mathbf{HH}^T\mathbf{H} + 2\lambda\mathbf{LH} + \alpha\mathbf{DE}_{k\times n}^T - 2\alpha\mathbf{DH} + 2\delta\mathbf{HE}_{k\times k} - 2\delta\mathbf{E}_{k\times n}^T.$$

Following the Karush-Kuhn-Tucker (KKT) conditions [5] that $\Phi \circledast \mathbf{H} = 0$, we have

$$(-4\mathbf{AH} + 4\mathbf{HH}^T\mathbf{H} + 2\lambda\mathbf{LH} + \alpha\mathbf{DE}_{k\times n}^T - 2\alpha\mathbf{DH} + 2\delta\mathbf{HE}_{k\times k} - 2\delta\mathbf{E}_{k\times n}^T)_{ij}h_{ij} = 0. \tag{6}$$

This is the fixed-point equation that the optimal solution must satisfy.

There are many ways to iteratively update h_{ij}. We here use the gradient descent method, i.e., $h_{ij} \leftarrow h_{ij} - \epsilon_{ij}\frac{\partial J}{\partial h_{ij}}$, and set $\epsilon_{ij} = \beta h_{ij}/(4\mathbf{HH}^T\mathbf{H} + 2\lambda\mathbf{S}'\mathbf{H} + \alpha\mathbf{DE}_{k\times n}^T + 2\delta\mathbf{HE}_{k\times k})_{ij}$, where β is a positive parameter in $(0, 1)$, leading to the following updating rule:

$$h_{ij} \leftarrow h_{ij}(1 - \beta + \beta\omega_{ij}), \tag{7}$$

where

$$\omega_{ij} = \left(\frac{4\mathbf{AH} + 2\lambda\mathbf{SH} + 2\alpha\mathbf{DH} + 2\delta\mathbf{E}_{k\times n}^T}{4\mathbf{HH}^T\mathbf{H} + 2\lambda\mathbf{S}'\mathbf{H} + \alpha\mathbf{DE}_{k\times n}^T + 2\delta\mathbf{HE}_{k\times k}}\right)_{ij}.$$

Due to the updating rule (7), $\mathbf{H}$ will eventually satisfy the fixed point equation of Equation (6) when converging. Clearly, $1 - \beta > 0$ and $\beta\omega_{ij} > 0$, therefore the updating rule (7) guarantees $\mathbf{H}$ to remain nonnegative in each iteration.

To conclude, Algorithm 3 summarizes the whole community detection algorithm based on NCNMF. Given a network $\mathcal{G}$, Algorithm 3 first extracts its adjacency matrix $\mathbf{A}$ and degree matrix $\mathbf{D}$ (line 1), and then gets the similarity matrix $\mathbf{S}$ via Algorithm 1 (line 2). Afterwards, Algorithm 3 optimizes NCNMF to get the community membership matrix $\mathbf{H}$ (lines 3-9). After NCNMF has been well optimized, Algorithm 3 loops over all nodes to divide $\mathcal{G}$ into disjoint communities according to the learned $\mathbf{H}$ (lines 10–12). Eventually, Algorithm 3 returns the detected communities (line 13).

5.2 Convergence Analysis

Before we proceed, we first state the following lemma:

Lemma 3. *Given parameters $\alpha > 0$ and $\delta > 0$, the function $f(\mathbf{H}) = \alpha\ \mathrm{Tr}[\mathbf{H}(\mathbf{E}_{k\times n} - \mathbf{H}^T)\mathbf{D}] + \delta\,||\mathbf{H1}_k - \mathbf{1}_n||_2^2$ is lower-bounded by*

$$n\delta - \frac{4nk\delta^2}{4\,(\delta k - (1 + k(k-1))\,\alpha n)},$$

under the conditions that

$$\delta > \frac{(1 + k(k-1))\,\alpha n}{k} \geq 0 \text{ and } \mathbf{H} \geq 0.$$

Proof. To begin with, we denote $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]^T$, where $\mathbf{h}_i \in \mathbb{R}_+^{k\times 1}$ is the transpose of the ith row vector of $\mathbf{H}$. We first row-wisely decompose $f(\mathbf{H}) = \sum_{i=1}^n f(\mathbf{h}_i)$, where

$$f(\mathbf{h}_i) = \alpha\sum_{j=1}^{k} d_{ii}h_{ij}(1 - h_{ij}) + \delta\left(\mathbf{h}_i^T\mathbf{1}_k - 1\right)^2 = \alpha d_{ii}(\mathbf{h}_i^T\mathbf{1}_k - \mathbf{h}_i^T\mathbf{h}_i) + \delta\left(\mathbf{h}_i^T\mathbf{1}_k - 1\right)^2.$$

Clearly, $f(\mathbf{h}_i)$ is a quadratic function in $\mathbf{h}_i$. It is computed that

$$f(\mathbf{h}_i) = \delta + (\alpha d_{ii}\mathbf{1}_k - 2\delta\mathbf{1}_k)^T\mathbf{h}_i + \frac{1}{2}\mathbf{h}_i^T\left(2\delta\mathbf{E}_{k\times k} - 2\alpha d_{ii}\mathbf{I}_k\right)\mathbf{h}_i.$$

The matrix $(2\delta\mathbf{E}_{k\times k} - 2\alpha d_{ii}\mathbf{I}_k)$ in the quadratic form is never positive semi-definite. Indeed, it has eigenvalues $2\delta k - 2\alpha d_{ii}, -2\alpha d_{ii}, \ldots, -2\alpha d_{ii}$. No matter how big δ is, there are always $k-1$ strictly negative eigenvalues of the same value $-2\alpha d_{ii}$. However, it is noted that, the normalized eigenvector associated with the largest eigenvalue is $\boldsymbol{\xi}_1 = [1, 1, \ldots, 1]/\sqrt{k} \in \mathbb{R}^k_{++}$.

Next, we can utilize the simultaneous nonnegativity of $\mathbf{h}_i$ and $\boldsymbol{\xi}_1$ to exploit deeper-level properties of the problem. We first expand $\mathbf{h}_i$ in terms of the bases of $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_k\}$, where $\{\boldsymbol{\xi}_2, \boldsymbol{\xi}_3, \ldots, \boldsymbol{\xi}_k\}$ are a system of orthonormal bases of $\text{span}(\{\boldsymbol{\xi}_1\})^\perp$ (s.t. $\text{span}(\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_k\}) = \mathbb{R}^k$ and all of them are orthonormal), as

$$\mathbf{h}_i = \beta_1\boldsymbol{\xi}_1 + \beta_2\boldsymbol{\xi}_2 + \ldots + \beta_k\boldsymbol{\xi}_k,$$

for which $\beta_1 = \mathbf{h}_i^T\boldsymbol{\xi}_1 \geq 0$ since $\mathbf{h}_i$ and $\boldsymbol{\xi}_1$ are both nonnegative. Equipped with this expansion, we can obtain

$$\begin{aligned}\frac{1}{2}\mathbf{h}_i^T\left(2\delta\mathbf{E}_{k\times k} - 2\alpha d_{ii}\mathbf{I}_k\right)\mathbf{h}_i =& \frac{1}{2}\left(\sum_{j=1}^k \beta_j\boldsymbol{\xi}_j\right)^T\left(2\delta\mathbf{E}_{k\times k} - 2\alpha d_{ii}\mathbf{I}_k\right)\left(\sum_{j=1}^k \beta_j\boldsymbol{\xi}_j\right)\\ =& \frac{1}{2}\sum_{j=1}^k \beta_j^2\left(\boldsymbol{\xi}_j^T(2\delta\mathbf{E}_{k\times k} - 2\alpha d_{ii}\mathbf{I}_k)\boldsymbol{\xi}_j\right).\end{aligned} \tag{8}$$

We here claim that $|\beta_j| \leq \sqrt{k}\beta_1,\ \forall\, j \neq 1$. This is because $\dim(\mathbb{R}^k) < \infty$ and thus

$$|\beta_j| = |\mathbf{h}_i^T\boldsymbol{\xi}_j| \leq \|\mathbf{h}_i\|_1\,\|\boldsymbol{\xi}_j\|_\infty \leq \|\mathbf{h}_i\|_1\,\|\boldsymbol{\xi}_j\|_2 = \|\mathbf{h}_i\|_1 = \mathbf{h}_i^T\mathbf{1}_k = \sqrt{k}\mathbf{h}_i^T\boldsymbol{\xi}_1 = \sqrt{k}\beta_1.$$

Therefore, by substituting the bounds for β_j's into Equation (8), we can obtain

$$\frac{1}{2}\mathbf{h}_i^T\left(2\delta\mathbf{E}_{k\times k} - 2\alpha d_{ii}\mathbf{I}_k\right)\mathbf{h}_i \geq \beta_1^2\left(\delta k - (1 + k(k-1))\,\alpha d_{ii}\right).$$

Besides, it is noted that

$$(\alpha d_{ii}\mathbf{1}_k - 2\delta\mathbf{1}_k)^T\mathbf{h}_i = \sqrt{k}(\alpha d_{ii}\boldsymbol{\xi}_1 - 2\delta\boldsymbol{\xi}_1)^T\mathbf{h}_i = \beta_1\sqrt{k}(\alpha d_{ii} - 2\delta).$$

Putting the above two formulas together, we obtain

$$f(\mathbf{h}_i) \geq \beta_1^2\left(\delta k - (1 + k(k-1))\,\alpha d_{ii}\right) + \beta_1\sqrt{k}(\alpha d_{ii} - 2\delta) + \delta.$$

Therefore, $f(\mathbf{h}_i)$ is lower-bounded if

$$\delta > \frac{(1 + k(k-1))\,\alpha n}{k} \geq \frac{(1 + k(k-1))\,\alpha d_{ii}}{k} \geq 0, \tag{9}$$

with the lower bound as

$$\delta - \frac{k(\alpha d_{ii} - 2\delta)^2}{4\left(\delta k - (1 + k(k-1))\,\alpha d_{ii}\right)} \geq \delta - \frac{4k\delta^2}{4\left(\delta k - (1 + k(k-1))\,\alpha n\right)}.$$

Hence, whenever δ satisfies Equation (9),

$$f(\mathbf{H}) = \sum_{i=1}^n f(\mathbf{h}_i) \geq n\delta - \frac{4nk\delta^2}{4\left(\delta k - (1 + k(k-1))\,\alpha n\right)},$$

as desired. □

The convergence of Algorithm 3 is theoretically guaranteed in the following theorem:

Theorem 4. *Given a small enough value of β and $\delta > (1 + k(k-1))\,\alpha n/k \geq 0$, Algorithm 3 will monotonically decrease the objective function of Problem (5) in each iteration, and converge eventually.*

Proof. By the completeness of real numbers, any bounded monotonic sequence must converge. As we use the gradient descent method to optimize Problem (5), whenever the learning rate is small (which is achieved by a small β), the objective function of Problem (5) will decrease monotonically. Besides, the objective function of Problem (5) has a lower bound, which is due to the fact that the first term is obviously nonnegative, the second term as the sum of several quadratic forms of the positive semi-definite matrix $\mathbf{L}$ is also nonnegative, while the last two terms are warranted to be lower-bounded by Lemma 3. Hence, Algorithm 3 converges. □

5.3 Complexity Analysis

Recall that n is the number of nodes, h is a constant which means that a node can only take at most h-order neighbors, k is the number of communities to be detected and t is the total number of iterations when Algorithm 3 converges. The computational complexity of Algorithm 3 consists of four parts:

(1) **Computing $\Gamma_p(v_i)$'s:** By our computing scheme, the computational complexity of computing $\Gamma_p(v_i)$'s for all $v_i \in \mathcal{V}$ and $p = 1, 2, \ldots, h$ is $O(n^2 + (h-1)n^3)$;
(2) **Constructing the similarity matrix:** There are a total of n nodes in the network, and computing the w-Jaccard similarity for each pair of nodes requires a time complexity of $O(h^2 n)$, thus the computational complexity of constructing the similarity matrix is $O(h^2 n^3)$;
(3) **Updating H:** The update of $\mathbf{H}$ consists of a lot of matrix multiplications and divisions, which takes $O(n^2 k + nk^2)$ time in each iteration. Since $k \leq n$, the total computational complexity of updating $\mathbf{H}$ is $O(tn^2 k)$;
(4) **Dividing $\mathcal{G}$:** For dividing $\mathcal{G}$ into communities, we only need to check the community membership vectors of all nodes. Therefore, the computational complexity of dividing $\mathcal{G}$ is $O(nk)$.

In summary, the overall computational complexity of Algorithm 3 is $O(n^2 + (h-1)n^3 + h^2 n^3 + tn^2 k + nk)$, which can be considered as $O(h^2 n^3 + tn^2 k)$, where h is a small constant in practice and usually less than or equal to 3.

6 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of our proposed method in comparison with several state-of-the-art community detection methods. Our algorithms are implemented in Python 3.7 and publicly available.[1] All experiments are conducted on a Ubuntu server with 3.70-GHz i9-10900K CPU and 128-GB main memory.

6.1 Experimental Settings

6.1.1 Networks. We use eight datasets of real-world networks with ground-truth communities as shown in Table 2. In each network dataset, communities are formed by the nodes within the same institution affiliation. As a preprocessing step, we remove all isolated nodes from the networks. The Texas, Cornell, Washington, and Wisconsin networks are downloaded from LINQS.[2] The Gene, Citeseer, Reality-call, and BZR networks are downloaded from Network Repository.[3]

6.1.2 Comparative Methods. We compare our algorithm NCNMF with thirteen state-of-the-art community detection methods, including PNMF [48], LPA [33], GNMF [6], SymmNMF [18], DeepWalk [30], ONMF [31], MNMF [41], Ego-Splitting [12], NNSED [38], DANMF [45], HPNMF [44], EdMot [21], and AGC [50]. For detailed descriptions of these methods, we refer the readers to

[1] See https://github.com/wowoHead/NCNMF.
[2] See https://linqs.soe.ucsc.edu/data.
[3] See https://networkrepository.com/index.php.

Table 2. Network Statistics of Eight Datasets

Statistic	Texas	Cornell	Washington	Wisconsin	Gene	Citeseer	Reality-call	BZR
# Nodes	185	195	217	262	1,103	3,264	6,809	14,479
# Edges	296	286	404	476	1,672	4,612	7,697	15,535
# Communities	5	5	5	5	2	6	2	10

Section 2. All these methods have already been shown to achieve good performance in community detection.

6.1.3 Parameter Settings. By default, we set the maximum number of iterations as 500 for all NMF-based methods. For each method, we run it 20 times and report the mean value with standard deviation. For our method NCNMF, we tune both λ and α in the range of $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$, set the stopping criterion as $(J_{t-1} - J_t)/J_{t-1} \leq 10^{-4}$, where J_t represents the objective function value in the tth iteration, and set the parameters $\delta = 10^5$, $h = 3$, $\Omega_{p,q} = (|\mathcal{V}|/|\mathcal{E}|)^{pq}$ [42], and $\beta = 0.5$ [11]. The PNMF, LPA, SymmNMF, DeepWalk, ONMF, MNMF, Ego-Splitting, NNSED, EdMot, and AGC are parameter-free methods, in which no parameter setting is needed. For GNMF, we set $\lambda = 100$ [6]. For HPNMF, we set $\lambda = 1$ and $\gamma = 10^{-2}$ [44], and for DANMF, we set the layer size as $n \rightarrow 256 \rightarrow 128 \rightarrow k$ and the maximum number of pre-training iterations as 100 [45]. This experiment aims at evaluating the accuracy of the discovered communities w.r.t. the ground-truth communities. Thus, for all competitive methods, we set the number of detected communities as the number of ground-truth communities, as listed in Table 2.

6.1.4 Evaluation Metrics. We adopt *F-score* [7] and *Accuracy* [26] to measure the quality of the detected communities of all algorithms, as all networks have ground-truth communities.

The *F-score* is defined as

$$F\text{-}score = \frac{2tp}{2tp + fp + fn},$$

where tp, fp, fn are the number of true positive, false positive and false negative hits obtained by the pair confusion matrix, respectively. The metric is bounded between 0 and 1. The larger the *F-score*, the better the community detection performance.

The *Accuracy* is defined as:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn},$$

where tp, tn, fp, fn are the number of true positive, true negative, false positive and false negative hits obtained by the pair confusion matrix, respectively. The metric is bounded between 0 and 1. The larger the *Accuracy*, the better the community detection performance.

6.2 Quality Evaluation of Community Detection Methods

This experiment evaluates the effectiveness of all community detection methods. Table 3 and Table 4 present the *F-score* and *Accuracy* performance of NCNMF and other state-of-the-art comparative methods on all the eight network datasets (parameters of NCNMF are tuned within the range introduced in Section 6.1, and the results under the optimal parameters are reported). We observe that NCNMF performs the best in 3/8 cases and 6/8 cases in terms of *F-score* and *Accuracy*, respectively, with comparison to thirteen state-of-the-art community detection methods. Although NCNMF is not the best in terms of *F-score* on the Texas, Cornell, Washington, and Citeseer networks, it achieves the second-best performance, and performs almost as well as the best methods. Besides, NCNMF both ranks at top-2 in 7/8 cases in terms of *F-score* and *Accuracy* performance. This outstanding performance strongly demonstrates the effectiveness of NCNMF. xvv

Table 3. The *F-score* Results of Different Community Methods on the Eight Networks

Method	Texas	Cornell	Washington	Wisconsin	Gene	Citeseer	Reality-call	BZR
PNMF	0.3830±0.0067	0.3137±0.0091	0.4090±0.0128	0.3298±0.0048	0.6184±0.0012	0.1750±0.0008	0.9870±0.0001	0.1581±0.0001
GNMF	**0.5761±0.0583**	0.3513±0.0391	0.4671±0.0398	0.3780±0.0387	0.6181±0.0019	0.3018±0.0030	0.9752±0.0036	0.1456±0.0001
LPA	0.3444±0.0001	0.2045±0.0001	0.2526±0.0001	0.2081±0.0001	0.0201±0.0001	0.0186±0.0001	0.0203±0.0001	0.0004±0.0001
SymmNMF	0.4016±0.0163	0.3146±0.0034	0.3193±0.0147	0.3204±0.0011	0.5029±0.0085	0.2920±0.0241	0.6887±0.0035	0.1615±0.0039
DeepWalk	0.3657±0.0287	0.2816±0.0057	0.3009±0.0131	0.2958±0.0073	0.5030±0.0028	0.2084±0.0091	0.6743±0.0043	0.1604±0.0052
ONMF	0.5322±0.0331	0.3302±0.0245	0.4458±0.0189	0.3567±0.0098	0.5742±0.0034	**0.3193**±0.0075	0.8106±0.0092	0.1618±0.0006
MNMF	0.2560±0.0438	0.2382±0.0391	0.2534±0.0262	0.2516±0.0412	0.5072±0.0316	0.1975±0.0296	0.6608±0.0341	0.1579±0.0175
NNSED	0.4780±0.0162	0.3375±0.0092	0.4641±0.0084	0.4099±0.0031	0.6020±0.0415	0.3108±0.0042	0.7325±0.0028	**0.5481**±0.0017
DANMF	0.5122±0.0128	0.3624±0.0095	0.4376±0.0142	0.3643±0.0089	0.6169±0.0212	0.2790±0.0009	0.8137±0.0015	0.2185±0.0203
Ego-Splitting	0.1686±0.0318	0.1586±0.0251	0.1315±0.0173	0.1320±0.0300	0.0650±0.0051	0.1029±0.0082	0.0392±0.0021	0.0048±0.0143
HPNMF	0.4284±0.0241	0.3307±0.0195	0.3175±0.0716	0.3339±0.0341	0.5665±0.0261	0.2339±0.0412	0.7353±0.0095	0.1615±0.0214
EdMot	0.1687±0.0010	0.1602±0.0042	0.1194±0.0173	0.1254±0.0076	0.0610±0.0062	0.0922±0.0127	0.0381±0.0197	0.0048±0.0005
AGC	0.5019±0.0001	**0.4370**±0.0129	**0.4966**±0.0001	0.4766±0.0229	0.6575±0.0140	0.2957±0.0023	0.9784±0.0001	0.3569±0.0259
NCNMF	0.5392±0.0161	0.4331±0.0015	0.4907±0.0115	**0.4784**±0.0006	**0.6700**±0.0003	0.3111±0.0057	**0.9871**±0.0005	0.1646±0.0049

Table 4. The *Accuracy* Results of Different Community Methods on the Eight Networks

Method	Texas	Cornell	Washington	Wisconsin	Gene	Citeseer	Reality-call	BZR
PNMF	0.2062±0.0310	0.1920±0.0358	0.2129±0.0298	0.2045±0.0226	0.2760±0.0181	0.1699±0.0107	0.0061±0.0010	0.1001±0.0031
GNMF	0.1845±0.0647	0.2228±0.0536	0.1762±0.0976	0.1917±0.0895	0.2788±0.0196	0.1719±0.0108	0.0064±0.0010	0.0998±0.0029
LPA	0.0486±0.0001	0.0951±0.0008	0.0691±0.0001	0.1870±0.0001	0.1922±0.0182	0.0018±0.0001	0.0048±0.0001	0.1006±0.0017
SymmNMF	0.3567±0.0481	0.1128±0.0248	0.1520±0.0391	0.3053±0.0481	0.2647±0.0217	0.1265±0.0291	0.0067±0.0548	0.1135±0.0174
DeepWalk	0.2064±0.0002	0.2099±0.0010	0.2105±0.0007	0.2206±0.0001	0.2718±0.0001	0.1687±0.0014	0.0066±0.0011	0.1012±0.0005
ONMF	0.1889±0.0516	0.1676±0.0251	0.2207±0.0816	0.1906±0.0913	0.2210±0.0418	0.1450±0.0754	0.0040±0.0328	0.0996±0.0515
MNMF	0.2054±0.0415	0.2153±0.0811	0.2672±0.0413	0.1908±0.0841	0.3055±0.0548	0.1734±0.0431	0.0060±0.0123	0.1002±0.0607
NNSED	0.0270±0.0041	0.0051±0.0012	0.0081±0.0003	0.0038±0.0005	0.1253±0.0026	0.1807±0.0105	0.0056±0.0001	**0.4662**±0.0204
DANMF	0.1189±0.0141	0.1487±0.0112	0.1889±0.0102	0.3854±0.0158	0.1577±0.0115	0.1274±0.0412	0.0084±0.006	0.1284±0.0109
Ego-Splitting	0.0594±0.0011	0.0717±0.0010	0.1105±0.0051	0.0496±0.0006	0.0181±0.0001	0.0171±0.0001	**0.0349**±0.0001	0.0022±0.0001
HPNMF	0.1997±0.0421	0.1894±0.0816	0.2016±0.0153	0.1755±0.0136	0.2697±0.0416	0.1445±0.0204	0.0070±0.0155	0.1022±0.0094
EdMot	0.0594±0.0017	0.0512±0.0021	0.0276±0.0009	0.0572±0.0017	0.0045±0.0002	0.0214±0.0010	0.0259±0.0017	0.0022±0.0006
AGC	0.1564±0.0878	0.4200±0.1154	0.1211±0.0275	0.1085±0.0287	0.0810±0.1353	0.1892±0.0130	0.0001±0.0028	0.1542±0.0580
NCNMF	**0.5421**±0.0154	**0.4387**±0.0210	**0.4926**±0.0312	**0.4562**±0.0088	**0.5468**±0.0071	**0.2878**±0.0410	0.0129±0.0001	0.1654±0.0018

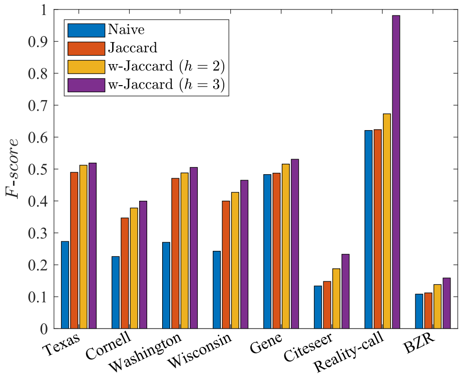


Fig. 4. Quality comparison of three similarity measures on the eight networks.

6.3 Similarity Measurement Comparison

In this experiment, we test the effectiveness of three similarity measures, including the Naive similarity, Jaccard similarity and our w-Jaccard similarity (when $h = 2$ and $h = 3$, respectively). Specifically, we run different versions of NCNMF equipped with different similarity measurements on all the networks, with $\lambda = 10^2$ and $\alpha = 1$. The *F-score* results are reported in Figure 4. We find that, on all the datasets, no matter the value of h is 2 or 3, the effects obtained by w-Jaccard are always better than those of the naive similarity and the Jaccard similarity. Especially, the *F-score* value corresponding to the w-Jaccard similarity ($h = 3$) on the Reality-call network is 0.9809, which is far larger than that of the naive and Jaccard similarity measures, which are 0.6209 and 0.6235, respectively. From this experiment, we can conclude that our new similarity measure has absolutely great advantages on all the datasets. Besides, we find that on small datasets (i.e., the

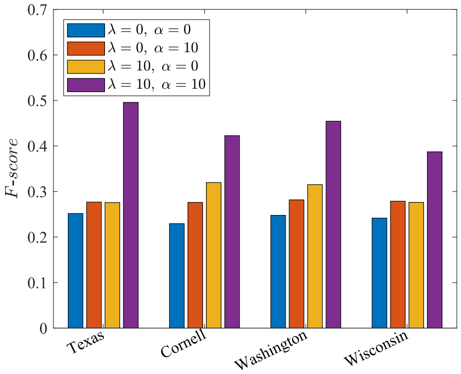


Fig. 5. Ablation study of NCNMF using different values of λ and α on the four networks.

Table 5. The Comparison Results of Running Time of Different Methods (in Seconds)

Network	PNMF	GNMF	LPA	SymmNMF	DeepWalk	ONMF	MNMF	NNSED	DANMF	Ego-Splitting	HPNMF	EdMot	AGC	NCNMF
Texas	0.086	0.423	0.152	0.082	0.524	0.731	0.902	0.167	3.154	0.045	0.250	0.016	0.572	0.245
Washington	0.108	0.491	0.097	0.056	0.617	0.424	3.714	0.974	6.741	0.060	0.187	0.028	0.641	0.287
Gene	12.293	3.601	1.019	0.151	6.410	3.065	2.613	3.015	15.751	0.364	1.638	0.210	7.194	4.479
Citeseer	113.600	13.816	1.295	0.990	12.020	479.100	12.320	17.091	517.380	1.491	3.446	0.525	17.195	16.816

Texas, Cornell, Washington, and Wisconsin networks), the w-Jaccard similarity obtains analogous performance no matter $h = 2$ or $h = 3$. This is because on small datasets, there may not be enough higher-order neighbors. But on larger datasets (i.e., the Citeseer and Reality-call networks), the w-Jaccard similarity with $h = 3$ shows better performance than that of $h = 2$, which verifies that the w-Jaccard similarity with $h = 3$ can obtain more useful information on large-scale networks.

6.4 Ablation Study

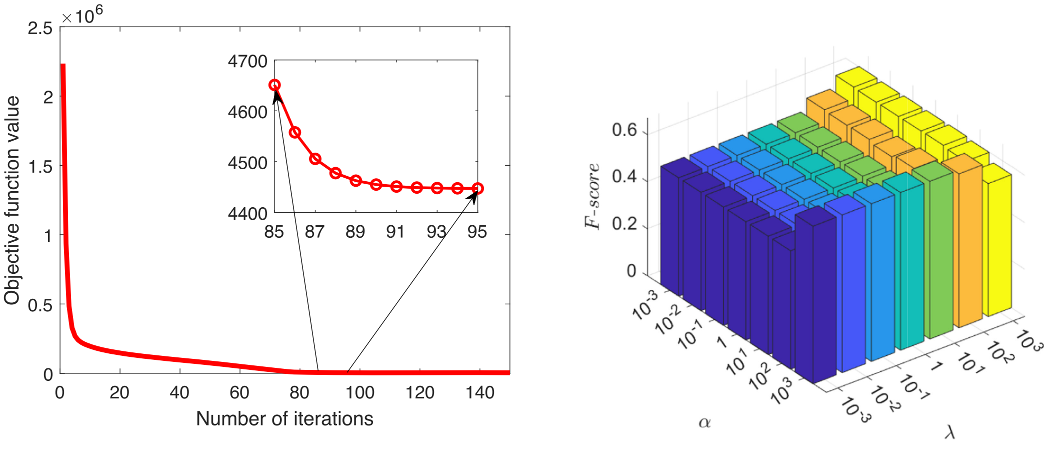
In this part, we present an ablation study to show the effectiveness of each component of NCNMF. Specifically, we run several ablated versions of NCNMF under different parameter settings, as introduced below:

(1) Setting $\lambda = 0$ and $\alpha = 0$, which shows the performance of the building block $\|\mathbf{A} - \mathbf{H}\mathbf{H}^T\|_F^2$.
(2) Setting $\lambda = 10$ and $\alpha = 0$, which shows the performance of the new similarity measure proposed in Section 4.1.
(3) Setting $\lambda = 0$ and $\alpha = 10$, which shows the performance of the sparse regularization proposed in Section 4.2.
(4) Setting $\lambda = 10$ and $\alpha = 10$, which shows the performance of fusing the new similarity measure and sparse regularization.

We report the *F-score* results of all variant NCNMF methods on the Texas, Cornell, Washington, and Wisconsin networks. Figure 5 shows that the combination of the new similarity measure and sparse regularization can yield better effects.

6.5 Efficiency Evaluation

In this experiment, we evaluate the efficiency of our proposed NCNMF method against other competitor methods. For NCNMF, we set parameters $\lambda = 10^2$ and $\alpha = 1$. Table 5 shows the comparison results of the running time of different community detection methods. As observed, NCNMF enjoys a relatively fast running time. Specifically, NCNMF runs faster than GNMF, DeepWalk, ONMF, MNMF, DANMF, HPNMF, and AGC on the Texas network, runs faster than GNMF,



(a) Convergence evaluation

(b) Sensitivity evaluation

Fig. 6. Convergence and sensitivity evaluations of NCNMF on the Citeseer network.

DeepWalk, ONMF, MNMF, NNSED, DANMF, and AGC on the Washington network, runs faster than PNMF, DeepWalk, DANMF, and AGC on the Gene network, and runs faster than PNMF, ONMF, NNSED, DANMF, and AGC on the Citeseer network. These evidences show the efficiency of NCNMF to some extent. Besides, due to a high cost of calculating our similarity matrix, NCNMF takes consistently more running time than four methods of LPA, SymmNMF, Ego-Splitting, and EdMot on these four datasets. Fortunately, this costly computation by NCNMF brings remarkable quality improvement compared to them. We conclude that NCNMF strikes a good balance between effectiveness and efficiency.

6.6 Convergence and Sensitivity Evaluations

In this part, we evaluate the convergence performance and parameter sensitivity for NCNMF. First, we run NCNMF on the Gene network with parameters $\lambda = 10^2$ and $\alpha = 1$ for convergence evaluation. From Figure 6(a), we observe that the objective function value decreases monotonically. That is, it drops rapidly at the very beginning, and then tends to achieve a convergence after 90 iterations, reflecting that Algorithm 3 converges in a very fast speed. It not only verifies the correctness of 4, but also fully demonstrates the efficiency of Algorithm 3 in community detection. Next, we run NCNMF on the same network with λ and α varying in the range of $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ for sensitivity analysis. From Figure 6(b), we can observe that NCNMF is relatively nonsensitive to λ and α. NCNMF has a good quality performance when $\alpha = 10^3$ or $10^2 \leq \lambda \leq 10^3$.

7 CONCLUSION

In this article, we have proposed a novel community detection method, NCNMF. Distinct from existing NMF-based community detection methods, NCNMF not only leverages h-order weighted Jaccard similarity to extract richer structural information, but also introduces the node centrality and *Gini* impurity to regularize the node-community memberships. To optimize NCNMF, we have derived an efficient algorithm, which can be theoretically guaranteed to converge. We have conducted extensive experiments to evaluate our NCNMF method on eight real-world benchmark networks, and experimental results have shown that NCNMF outperforms the state-of-the-arts markedly. For future work, we plan to study how to extend NCNMF to directed networks and detect overlapping communities.

REFERENCES

[1] Lada A. Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social Networks* 25, 3 (2003), 211–230.

[2] Ethem Alpaydin. 2020. *Introduction to Machine Learning*. MIT Press, Massachusetts.

[3] Kamal Berahmand, Asgarali Bouyer, and Mahdi Vasighi. 2018. Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes. *IEEE Transactions on Computational Social Systems* 5, 4 (2018), 1021–1033.

[4] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), 10008.

[5] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, New York.

[6] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang. 2010. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8 (2010), 1548–1560.

[7] Tanmoy Chakraborty, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly. 2017. Metrics for community analysis: A survey. *Comput. Surveys* 50, 4 (2017), 1–37.

[8] Qiong Chen, Ting-Ting Wu, and Ming Fang. 2013. Detecting local community structures in complex networks based on local degree central nodes. *Physica A: Statistical Mechanics and its Applications* 392, 3 (2013), 529–537.

[9] Aaron Clauset, Mark E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical Review E* 70, 6 (2004), 066111.

[10] Aniello De Santo, Antonio Galli, Vincenzo Moscato, and Giancarlo Sperlì. 2021. A deep learning approach for semi-supervised community detection in online social networks. *Knowledge-Based Systems* 229 (2021), 107345.

[11] Chris Ding, Xiaofeng He, and Horst D. Simon. 2005. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 5th SIAM International Conference on Data Mining*. 606–610.

[12] Alessandro Epasto, Silvio Lattanzi, and Renato Paes Leme. 2017. Ego-splitting framework: From non-overlapping to overlapping clusters. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 145–154.

[13] Giorgio Fagiolo. 2007. Clustering in complex directed networks. *Physical Review E* 76, 2 (2007), 026107.

[14] Michelle Girvan and Mark E. J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99, 12 (2002), 7821–7826.

[15] Ling Huang, Chang-Dong Wang, and S. Yu Philip. 2021. Higher order connection enhanced community detection in adversarial multiview networks. *IEEE Transactions on Cybernetics* 29, 6 (2021), 1–15.

[16] Hong Jin, Wei Yu, and ShiJun Li. 2019. Graph regularized nonnegative matrix tri-factorization for overlapping community detection. *Physica A: Statistical Mechanics and its Applications* 515 (2019), 376–387.

[17] Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 1 (1953), 39–43.

[18] Da Kuang, Chris Ding, and Haesun Park. 2012. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 12th SIAM International Conference on Data Mining*. 106–117.

[19] Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by nonnegative matrix factorization. *Nature* 401, 6755 (1999), 788–791.

[20] Alberto Leon-Garcia and Indra Widjaja. 2003. *Communication Networks*. McGraw-Hill, Inc.

[21] Pei-Zhen Li, Ling Huang, Chang-Dong Wang, and Jian-Huang Lai. 2019. Edmot: An edge enhancement approach for motif-aware community detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 479–487.

[22] Ye Li, Chaofeng Sha, Xin Huang, and Yanchun Zhang. 2018. Community detection in attributed graphs: An embedding approach. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[23] Xiao Liu, Wenjun Wang, Dongxiao He, Pengfei Jiao, Di Jin, and Carlo Vittorio Cannistraci. 2017. Semi-supervised community detection based on nonnegative matrix factorization with node popularity. *Information Sciences* 381 (2017), 304–321.

[24] Chao Lyu, Yuhui Shi, and Lijun Sun. 2019. A novel local community detection method using evolutionary computation. *IEEE Transactions on Cybernetics* 51, 6 (2019), 3348–3360.

[25] Xiaoke Ma and Di Dong. 2017. Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering* 29, 5 (2017), 1045–1058.

[26] Charles E. Metz. 1978. Basic principles of ROC analysis. In *Seminars in Nuclear Medicine*, Vol. 8. Elsevier, 283–298.

[27] Mark E. J. Newman. 2001. Clustering and preferential attachment in growing networks. *Physical Review E* 64, 2 (2001), 025102.

[28] Mark E. J. Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69, 2 (2004), 026113.

[29] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435, 7043 (2005), 814–818.

[30] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 701–710.

[31] Filippo Pompili, Nicolas Gillis, P.-A. Absil, and François Glineur. 2014. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing* 141 (2014), 15–25.

[32] Ioannis Psorakis, Stephen Roberts, Mark Ebden, and Ben Sheldon. 2011. Overlapping community detection using Bayesian nonnegative matrix factorization. *Physical Review E* 83, 6 (2011), 066114.

[33] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 76, 3 (2007), 036106.

[34] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. 2019. Gemsec: Graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 65–72.

[35] Farshad Saberi-Movahed, Mahyar Mohammadifard, Adel Mehrpooya, Mohammad Rezaei-Ravari, Kamal Berahmand, Mehrdad Rostami, Saeed Karami, Mohammad Najafzadeh, Davood Hajinezhad, Mina Jamshidi, Farshid Abedi, Mahtab Mohammadifard, Elnaz Farbod, Farinaz Safavi, Mohammadreza Dorvash, Negar Mottaghi-Dastjerdi, Shahrzad Vahedi, Mahdi Eftekhari, Farid Saberi-Movahed, Hamid Alinejad-Rokny, Shahab S. Band, and Iman Tavassoly. 2022. Decoding clinical biomarker space of COVID-19: Exploring matrix factorization-based feature selection methods. *Computers in Biology and Medicine* 146 (2022), 105426.

[36] Junming Shao, Zhichao Han, Qinli Yang, and Tao Zhou. 2015. Community detection based on distance dynamics. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1075–1084.

[37] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Di Jin, Quan Z. Sheng, and Philip S. Yu. 2022. A comprehensive survey on community detection with deep learning. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[38] Bing-Jie Sun, Huawei Shen, Jinhua Gao, Wentao Ouyang, and Xueqi Cheng. 2017. A nonnegative symmetric encoder-decoder approach for community detection. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*. 597–606.

[39] Xiangyi Teng, Jing Liu, and Mingming Li. 2019. Overlapping community detection in directed and undirected attributed networks using a multi-objective evolutionary algorithm. *IEEE Transactions on Cybernetics* 51, 1 (2019), 138–150.

[40] René Vidal. 2011. Subspace clustering. *IEEE Signal Processing Magazine* 28, 2 (2011), 52–68.

[41] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community preserving network embedding. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 203–209.

[42] Jianshe Wu, Yunting Hou, Yang Jiao, Yong Li, Xiaoxiao Li, and Licheng Jiao. 2015. Density shrinking algorithm for community detection with path based similarity. *Physica A: Statistical Mechanics and its Applications* 433, 6 (2015), 218–228.

[43] Wenhui Wu, Yuheng Jia, Sam Kwong, and Junhui Hou. 2018. Pairwise constraint propagation-induced symmetric nonnegative matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems* 29, 12 (2018), 6348–6361.

[44] Fanghua Ye, Chuan Chen, Zhiyuan Wen, Zibin Zheng, Wuhui Chen, and Yuren Zhou. 2019. Homophily preserving community detection. *IEEE Transactions on Neural Networks and Learning Systems* 31, 8 (2019), 2903–2915.

[45] Fanghua Ye, Chuan Chen, and Zibin Zheng. 2018. Deep autoencoder-like nonnegative matrix factorization for community detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1393–1402.

[46] Fanghua Ye, Chuan Chen, Zibin Zheng, Rong-Hua Li, and Jeffrey Xu Yu. 2019. Discrete overlapping community detection with pseudo supervision. In *Proceedings of the 19th IEEE International Conference on Data Mining*. 708–717.

[47] Fanghua Ye, Shenghui Li, Zhiwei Lin, Chuan Chen, and Zibin Zheng. 2018. Adaptive affinity learning for accurate community detection. In *Proceedings of the 18th IEEE International Conference on Data Mining*. 1374–1379.

[48] Zhijian Yuan and Erkki Oja. 2005. Projective nonnegative matrix factorization for image compression and feature extraction. In *Scandinavian Conference on Image Analysis*. 333–342.

[49] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. 2014. *Social Media Mining: An Introduction*. Cambridge University Press, New York.

[50] Xiaotong Zhang, Han Liu, Qimai Li, and Xiao-Ming Wu. 2019. Attributed graph clustering via adaptive graph convolution. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 4327–4333.

[51] Yu Zhang and Dit-Yan Yeung. 2012. Overlapping community detection via bounded nonnegative matrix tri-factorization. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 606–614.

[52] Zhongying Zhao, Hui Zhou, Liang Qi, Liang Chang, and MengChu Zhou. 2021. Inductive representation learning via CNN for partially-unseen attributed networks. *IEEE Transactions on Network Science and Engineering* 8, 1 (2021), 695–706.

Received 10 June 2022; revised 8 December 2022; accepted 18 December 2022