A Unified Semi-Supervised Community Detection Framework Using Latent Space Graph Regularization

Liang Yang, Xiaochun Cao, *Senior Member, IEEE*, Di Jin, Xiao Wang, and Dan Meng, *Member, IEEE*

***Abstract*—Community structure is one of the most important properties of complex networks and is a foundational concept in exploring and understanding networks. In real world, topology information alone is often inadequate to accurately find community structure due to its sparsity and noises. However, potential useful prior information can be obtained from domain knowledge in many applications. Thus, how to improve the community detection performance by combining network topology with prior information becomes an interesting and challenging problem. Previous efforts on utilizing such priors are either dedicated or insufficient. In this paper, we firstly present a unified interpretation to a group of existing community detection methods. And then based on this interpretation, we propose a unified semi-supervised framework to integrate network topology with prior information for community detection. If the prior information indicates that some nodes belong to the same community, we encode it by adding a graph regularization term to penalize the latent space dissimilarity of these nodes. This framework can be applied to many widely-used matrix-based community detection methods satisfying our interpretation, such as nonnegative matrix factorization, spectral clustering, and their variants. Extensive experiments on both synthetic and real networks show that the proposed framework significantly improves the accuracy of community detection, especially on networks with unclear structures.**

***Index Terms*—Community detection, graph regularization, nonnegative matrix factorization (NMF), semi-supervised framework, spectral clustering (SC).**

Manuscript received June 11, 2014; revised October 17, 2014; accepted November 23, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61422213, Grant 61332012, and Grant 61303110, in part by the National Basic Research Program of China under Grant 2013CB329305, in part by the 100 Talents Programme of the Chinese Academy of Sciences, in part by the Ph.D. Programs Foundation of Ministry of Education of China under Grant 20130032120043, in part by the Open Project Program of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education of China under Grant 93K172013K02, and in part by the National Training Programs of Innovation and Entrepreneurship for Undergraduates under Grant 201410069040. This paper was recommended by Associate Editor Y. Jin.

L. Yang is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and also with the School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China.

X. Cao is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: caoxiaochun@iie.ac.cn).

D. Jin and X. Wang are with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: jindi@tju.edu.cn).

D. Meng is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2377154

I. Introduction

NETWORKS have become ubiquitous in real life. In many different disciplines, data exist in the form of networks, such as social networks [1], [2], biological networks [1], [3], and technological networks [4]. The area of network analysis has attracted many researchers from different fields such as physics, biology, mathematics, and computer science. Networks can be modeled as graphs by regarding each entity as a vertex and each link as an edge. It has been shown that networks have a structure of modules or communities which are subgraphs whose vertices are more tightly connected with each other than with vertices outside the subgraph [5], [6]. For example, a set of papers which cite much more papers in their own field than other fields can be regarded as a community. Although there is no general and widely-accepted definition of community structure due to the variation of applications, community structure is one of the most important properties of networks and is the foundational concept in exploring and understanding them.

Although a large number of community detection algorithms have been proposed [7], [8], most of them only take into account the topology information, and regard a community as a set of nodes which have similar link-pattern [9], [10]. These kinds of methods work well in the network with clear structure, i.e., the amount of intracommunity connections is much larger than that of the intercommunity connections. But they will degrade or fail when nodes have a large amount of connections to nodes in other communities. In fact, community detection is not just a graph partition task which ignores node's meaning, but a semantic clustering problem [11]. Specifically, link information alone is inadequate to accurately determine community structure for two reasons. Firstly, due to the complexity of network structure, such as overlapping communities or hierarchical structures, many traditional methods will degrade when community structure is not clear. For example, the network of U.S. political books, shown in Fig. 11, only has two densely connected communities. However, the real number of communities is three [12] since the third community is the overlapping part of the two communities as we observed. Secondly, it is hard to accurately detect the community due to the sparsity of the topology information. Recent research [13], [14] shows that there is a phase transition threshold on the difference between intra and inter community edge number, below which communities are impossible for any algorithm to detect, which is named as community detectability. Many community detection algorithms, such as spectral methods, succeed when the network is sufficiently dense [14], and they fail significantly when one gradually increases the number of external edges between communities [13], [15]. And this phenomenon has been found in heterogeneous networks [16] and interconnected networks [17].

In many real scenarios, some prior information, which is often in the form of pairwise constraints, is available for community detection. Currently, several methods have been proposed to make use of this type of information [18]–[23], which are named as semi-supervised community detection. In additional, it is demonstrated that the accuracy and robustness will be significantly improved only with limited prior information, especially on the most real-world complicated and noisy networks. It is obvious that, therefore, the process of community detection will be benefited when we incorporate all the information available from different sources, which includes not only the information of networks topology, but also the prior information. For example, if we have found the keywords or tf-idf feature of papers or web pages are similar or they are written by the same authors or under the same domain name, we can make use of these useful information as prior to guide the community detection algorithms by putting them into the same community, even though they may not cite or link with each other in network topology.

In this paper, we first present a unified interpretation under which a group of widely-used matrix based community detection algorithms can be analyzed, including nonnegative matrix factorization (NMF), spectral clustering (SC), and their variants. The inputs of these community detection algorithms are the adjacency matrices which represent the network topology information. And each row or column can be regarded as the feature or property representation of the corresponding node. They first obtain new property representations in latent space for each node by optimizing different objective functions, and then clustering nodes in that latent space. For example, NMF algorithms obtain the new property representation by factorizing the adjacency matrix into two nonnegative matrices. For another example, SC algorithms obtain it by finding the meaningful eigenvectors. Thus we give this type of methods a unified interpretation, i.e., clustering in the latent space.

And then based on this interpretation we propose a unified semi-supervised community detection framework for these methods which not only combines prior information with topology information, but also balances them to improve the performance of community detection. Under our interpretation, nodes are clustered based on the new property representation. It is believed that the property representation of intracommunity nodes are more similar than that of intercommunity nodes. If we have priors that some nodes belong to the same community, we introduce a graph regularization term to incorporate the prior information into the original objective functions. Besides, the proposed framework is flexible to balance the factors between the topology information and prior information according to reliability of priors. Furthermore, because we characterize different constraints using different terms in the objective function, we treat semi-supervised community detection as a brand new problem instead of handling it as a preprocess of traditional community detection problem.

We summarize the main contributions as follows.

1) We present a unified interpretation, i.e., clustering in the latent space, to a group of widely-used community detection algorithms, including NMF, SC, and their variants.
2) Based on the unified interpretation, we propose a general semi-supervised community detection framework which fully utilizes the must-link priors. Most importantly, we treat semi-supervised community detection problem as a brand new problem instead of handling it as a preprocess of traditional community detection problem.
3) Under the proposed framework, we give a formal analysis of the impacts of the topology information and prior information, and show how to balance them to improve the performance on different networks.

The remainder of this paper is organized as follows. A brief review of the related works on community detection is given in Section II. Section III introduces a unified interpretation to a group of existing community detection algorithms, and Section IV describes our graph regularized semi-supervised framework in details. We also provide corresponding algorithms and analyze the computational complexity. Extensive experiments on synthetic and real datasets are presented in Section V. Finally, Section VI concludes this paper by highlighting our main contributions.

II. Related Work

In the past few years, a large number of community detection algorithms have been proposed and some of them have achieved good performance in many fields [7], [8], [24]–[27]. These algorithms can be divided into several categories: divisive algorithms, e.g., GN algorithm proposed by Girvan and Newman [1]; modularity optimization methods e.g., FN algorithm [6], extremal optimization [28], and spectral optimization [29]; overlapping methods, e.g., Clique percolation method [24]; multiobjective community detection [30]; and some methods on dynamic networks [31] and multislice networks [32]. Most of these methods, however, only take into account the topology information but ignore the prior information. They work well in the network with clear structure, but degrade or fail when network structure is vague to detect due to the phase transition phenomenon [13], [14]. In real world, link information alone is inadequate to accurately determine community structure for the complexity and sparsity of the networks.

Recently, there are many semi-supervised community detection algorithms be proposed [18]–[23]. Eaton and Mansbach [18] employed a spin-glass model from statistical physics, which is a generalization of modularity Q function, to combine external knowledge into the community detection process. Ver Steeg *et al.* [19] examined the impact of pairwise constraints on the clustering accuracy from the viewpoint of statistical mechanics. However, they only focus on a random network composed of two equal-sized clusters. Allahverdyan *et al.* [20] studied the problem of semi-supervised graph clustering by integrating known cluster assignments for a fraction of nodes. By proving the equivalence of modularity density function [33] and symmetric NMF (SNMF), Ma *et al.* [21] encoded the must-link and cannot-link constraints into the adjacency matrix and factorize it to get the indicator matrix. Zhang [22] extends this method to other community detection algorithms besides SNMF. He directly modifies the adjacency matrix, which is equivalent to connect and disconnect edges between must-link and cannot-link pairs. Later, Zhang *et al.* [23] extended [22] by adding a logical inference step to better utilize the two types of prior information. It is worth noting that, all these semi-supervised community detection methods encode the labeled data by transferring the prior information into the topology information and modifying the adjacency matrix directly. Although this is the easiest and most straightforward way to use prior information, there exists a main drawback, i.e., directly connecting the nodes belonging to the same community cannot guarantee that they are classified into the same community. In other words, they do not fully utilize the must-link priors. By modifying network topology, existing methods often transfer the semi-supervised detection into the traditional (unsupervised) detection. They only take it as a preprocess problem of community detection and cannot describe the natural properties of the original semi-supervised problem.

III. Preliminary and Unified Interpretation

A network can be modeled as a graph $G = (V, E)$, in which V is the set of the vertices, and E is the set of the edges each of which connects two vertices in V. For simplicity, we assume G is a undirected and unweighted graph which contains N vertices as shown in Fig. 1. We make use of a nonnegative symmetric binary matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}_+^{N\times N}$ to denote the adjacency matrix of G in which entry a_{ij} denotes whether there is an edge between vertices i and j. $a_{ij} = 1$ if and only if there is an edge between vertices i and j, and $a_{ij} = 0$ otherwise. And we define $a_{ii} = 0$ for all $1 \le i \le N$. Furthermore, we assume there are K communities in the network, which is known as a prior. Community detection problem is to divide these nodes into K different groups based on the topology information, $\mathbf{A}$. We first briefly introduce two representative examples, i.e., NMF in Section III-A and SC in Section III-B, and then give a unified interpretation to these kinds of algorithms in Section III-C.

A. NMF in Community Detection

In the generative process of the network discussed in [9], a_{ij} is the observed variable, which denotes the probability of interactions between vertices i and j. We assume the probability of existing a connection between vertices i and j is determined by the probability that they generate in-edge and out-edge which belong the same community. We define two latent variables $\mathbf{W} = [w_{ik}] \in \mathbb{R}_+^{N\times K}$ and $\mathbf{H} = [h_{jk}] \in \mathbb{R}_+^{N\times K}$ whose elements w_{ik} and h_{ik} represent the probability that node i generates an in-edge and an out-edge that belong to the community k, respectively. These latent variables also imply the probability that node i belongs to the in- or out- community k. Each row of $\mathbf{W}$ or $\mathbf{H}$ can be seen as the membership distribution of one vertex as shown in Fig. 1. So the probability that vertices i and j connect with each other is expressed as

$$\hat{a}_{ij} = \sum_{k=1}^{K} w_{ik} h_{jk}. \tag{1}$$

As a result, we transform community detection problem to the NMF problem $\hat{\mathbf{A}} = \mathbf{W}\mathbf{H}^T$. And in each row of $\mathbf{W}$ or $\mathbf{H}$, the index of the largest element is the community. The adjacency matrix $\mathbf{A}$ is asymmetric if the network is directed, while the adjacency matrix $\mathbf{A}$ is symmetric and the factorized $\mathbf{W}$ and $\mathbf{H}$ differ by a constant multiplier if the network is undirected. In this paper, we focus on undirected unweight networks and use $\mathbf{H}$ to decide the nodes' membership. From the viewpoint of clustering, we can regard the factorization process as projecting the N dimension feature in the adjacent matrix into a K dimension latent space.

There are two common objective (loss) functions that quantify the quality of the factorization result. The first is based on the square loss function [10], [34] which is equivalent to the square of the Frobenius norm of the difference between two matrices

$$\mathcal{L}_{\text{LSE}}\left(\mathbf{A}, \mathbf{W}\mathbf{H}^T\right) = \left|\left|\mathbf{A} - \mathbf{W}\mathbf{H}^T\right|\right|_F^2. \tag{2}$$

And the second is based on the Kullback–Leibleer divergence (KL-divergence) between two matrices

$$\mathcal{L}_{\text{KL}}\left(\mathbf{A}, \mathbf{W}\mathbf{H}^T\right) = \text{KL}\left(\mathbf{A}||\mathbf{W}\mathbf{H}^T\right). \tag{3}$$

Various applications prefer different types of loss functions.

One variant of NMF is SNMF [35] which introduces the symmetry constraints into the NMF framework. If the network is undirected, the adjacency matrix is symmetric. So the factorization should be symmetric as follows:

$$\mathcal{L}_{\text{SYM}}\left(\mathbf{A}, \mathbf{H}\mathbf{H}^T\right) = \left|\left|\mathbf{A} - \mathbf{H}\mathbf{H}^T\right|\right|_F^2. \tag{4}$$

Since $\mathcal{L}_{\text{LSE}}$, $\mathcal{L}_{\text{KL}}$ and $\mathcal{L}_{\text{SYM}}$ are not convex in both $\mathbf{W}$ and $\mathbf{H}$, it is not easy to develop algorithms to find the global minimum of these loss functions. But since they are convex in either $\mathbf{W}$ or $\mathbf{H}$, Lee and Seung [36] presented iterative updating algorithms to minimize the objective function $\mathcal{L}_{\text{LSE}}$ as

$$w_{ik} \leftarrow w_{ik}\frac{(\mathbf{A}\mathbf{H})_{ik}}{\left(\mathbf{W}\mathbf{H}^T\mathbf{H}\right)_{ik}}, \quad h_{jk} \leftarrow h_{jk}\frac{\left(\mathbf{A}^T\mathbf{W}\right)_{jk}}{\left(\mathbf{H}\mathbf{W}^T\mathbf{W}\right)_{jk}}. \tag{5}$$

Similarly, $\mathcal{L}_{\text{KL}}$ in (3) and $\mathcal{L}_{\text{SYM}}$ in (4) can be solved through following two updating schemes, respectively:

$$\begin{aligned} w_{ik} &\leftarrow w_{ik}\frac{\sum_j \left(a_{ij}h_{jk}/\sum_k w_{ik}h_{jk}\right)}{\sum_j h_{jk}} \\ h_{jk} &\leftarrow h_{jk}\frac{\sum_i \left(a_{ij}w_{ik}/\sum_k w_{ik}h_{jk}\right)}{\sum\limits_i w_{ik}} \end{aligned} \tag{6}$$

$$h_{ik} \leftarrow h_{ik}\left(\frac{1}{2} + \frac{(\mathbf{A}\mathbf{H})_{ik}}{\left(2\mathbf{H}\mathbf{H}^T\mathbf{H}\right)_{ik}}\right). \tag{7}$$

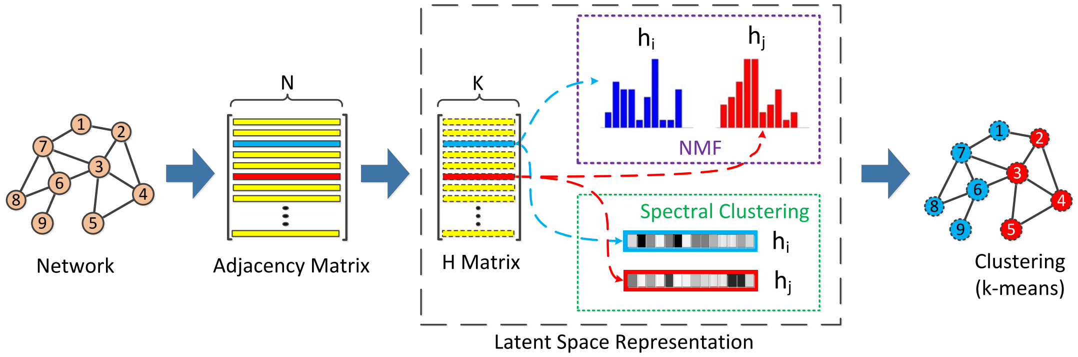


Fig. 1. Unified interpretation of some existing community detection algorithms. Although they have different formulations and meanings, these algorithms can be interpreted as follows. First, we take the adjacency matrix which encodes the topology information as input. Next, we obtain the new property representation. Finally, clustering is based on the new property representation and different distance metrics. For example, NMF obtains the new property representation by factorizing the adjacency matrix into two nonnegative ones.

B. SC in Community Detection

Different from NMF which obtains the formulation from the generative process of networks, SC is introduced to community detection by Newman [6], [29] to maximize modularity Q. The Q is defined as the difference between the number of edges within communities and the expected number of such edges over all pairs of vertices

$$Q = \frac{1}{4m}\sum_{ij}\left(a_{ij} - \frac{k_i k_j}{2m}\right)\left(h_i h_j\right) \tag{8}$$

where the network has two communities. h_i equals to 1 (0) if vertex i belongs to the first (second) group, $k_i k_j/2m$ is the expected number of edges between vertices i and j if edges are placed randomly. Here k_i is the degree of vertex i and $m = 1/2\sum_i k_i$ is the total number of edges in the network. By defining modularity matrix $\mathbf{B} = [b_{ij}] \in \mathbb{R}^{N\times N}$ whose elements are $b_{ij} = a_{ij} - k_i k_j/2m$, modularity Q can be written as

$$Q = \frac{1}{4m}\mathbf{h}^T\mathbf{B}\mathbf{h} \tag{9}$$

where $\mathbf{h} = [h_i] \in \mathbb{R}^N$ is an indicator vector. Maximizing (9) has been proved to be a NP-hard problem [37], and there are many optimization algorithms be proposed, such as extremal optimization [28]. In practice, we can relax the problem by allowing variable h_i to take any real value between -1 and 1, i.e., $\mathbf{h}^T\mathbf{h} = 1$.To generalize the formulations in (9) to $K > 2$ communities, by defining an indicator matrix $\mathbf{H} = [h_{ij}] \in \mathbb{R}^{N\times K}$ we can obtain

$$\begin{aligned}&\mathcal{L}_{\text{MOD}}(\mathbf{H},\mathbf{B}) = Q = \text{Tr}\left(\mathbf{H}^T\mathbf{B}\mathbf{H}\right)\\&\text{s.t. Tr}\left(\mathbf{H}^T\mathbf{H}\right) = N\end{aligned} \tag{10}$$

where $\text{Tr}(\cdot)$ denotes the trace of a matrix. Based on Rayleigh quotient, the solution to this problem is the largest K eigenvectors of the modularity matrix $\mathbf{B}$. The index of the largest element in each row of $\mathbf{H}$ indicates the community which the node belongs to. This $\mathbf{H}$ is similar to that in (2). Differently, researchers along this line use more flexible clustering methods, such as k-means. As discussed in [38], besides the modularity matrix $\mathbf{B}$ ($\mathcal{L}_{\text{MOD}}$), spectral analysis achieves great success in uncovering the community structure based on the adjacency matrix $\mathbf{A}$ ($\mathcal{L}_{\text{ADJ}}$), standard Laplacian matrix $\mathbf{L}_S = \mathbf{D} - \mathbf{A}$ ($\mathcal{L}_{\text{LAP}}$), and normalized Laplacian matrix $\mathbf{L}_N = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ ($\mathcal{L}_{\text{NLAP}}$), where $\mathbf{D}$ is a diagonal matrix with the ith diagonal element, which is the degree of node i, i.e., k_i.

C. Unified Interpretation

Although why NMF and SC make sense and how they work in community detection are very different, these types of algorithms can be general interpreted from the viewpoint of clustering. The input of community detection algorithms is the topology information of the network, which can be represented as the adjacent matrix $\mathbf{A}$. Many community detection algorithms first obtain a new matrix from adjacent matrix by minimizing an objective function. The new matrix can be regarded as a representation in the latent space. The algorithms then divide nodes by clustering rows in the new matrix using k-means or other clustering algorithms. Thus we can summarize these methods as the process of clustering in the latent space, which is shown in Fig. 1. Besides, there are many other algorithms, such as Markov clustering [39], which follow our interpretation.

IV. Our Framework

In this section, based on the unified interpretation discussed above, we first present the unified semi-supervised community detection framework using latent space graph regularization and discuss how it can be used to NMF, SC, and their variants in Section IV-A. And then we present some specific solutions to these optimization problems in Section IV-B. Finally, the complexity analysis and model selection are offered in Sections IV-C and IV-D.

A. Overview

In this section, we propose the unified graph regularized semi-supervised framework which can make use of the prior information to improve the performance of community detection. The main idea is displayed in Fig. 2. Recall that many community detection algorithms can be generally interpret as the process of clustering in the latent space. It is easy to find that the nodes belonging to the same community should have similar representations in latent space. If we have known that vertices i and j belong to the same community, then $\mathbf{h}_i$ and $\mathbf{h}_j$, which are the ith and jth rows in the indicator matrices $\mathbf{H}$

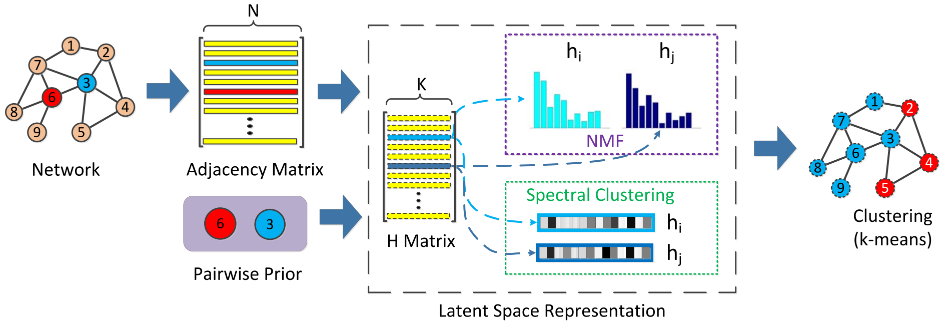


Fig. 2. Proposed semi-supervised framework which combines pairwise prior information with topology information. Based on the unified interpretation shown in Fig. 1, to enforce two nodes to be classified into the same community, the natural way is to let them have similar property representations. If we have known that nodes 3 and 9 belong to the same community, we can minimize the difference between two new property representations, i.e., $\mathbf{h}_3$ and $\mathbf{h}_9$, by adding a graph regularization term in the original objective function. Thus in our semi-supervised framework, the node property representation, which is used to cluster, is determined by both topology information and pairwise prior information.

in (2)–(4) and (10), should be similar. Therefore by minimizing the difference between these two rows, we can assign them into the same community.

To measure the similarity of the two vectors which denote the new representations of vertices i and j, we can use either square distance or KL-divergence

$$\mathcal{D}_{\text{LSE}}\left(\mathbf{h}_i, \mathbf{h}_j\right) = \left|\left|\mathbf{h}_i - \mathbf{h}_j\right|\right|_2^2 = \sum_{k=1}^{K}\left(h_{ik} - h_{jk}\right)^2 \qquad (11)$$

$$\mathcal{D}_{\text{KL}}\left(\mathbf{h}_i || \mathbf{h}_j\right) = \sum_{k=1}^{K}\left(h_{ik}\log\left(\frac{h_{ik}}{h_{jk}}\right) - h_{ik} + h_{jk}\right). \qquad (12)$$

There is no clear answer to the question that which distance is better, which depends on the specific applications.

If we have priors that some nodes belong to the same community, we can formulate these information with a collection of triples (i, j, o_{ij}) which means nodes i and j belong to the same community with reliability o_{ij} ($o_{ij} = 0$ if we do not have any prior information about the relationship between i and j). We express collection as $C = \{(i, j, o_{ij})\}_{i,j=1}^{N}$.

The constraints from these priors can be formulated as

$$\begin{aligned}\mathcal{R}_{\text{LSE}}\left(\{o_{ij}\}, \mathbf{H}\right) &= \frac{1}{2}\sum_{(i,j,o_{ij})\in C} o_{ij}\mathcal{D}_{\text{LSE}}\left(\mathbf{h}_i, \mathbf{h}_j\right) \\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} o_{ij}\left|\left|\mathbf{h}_i - \mathbf{h}_j\right|\right|_2^2. \qquad (13)\end{aligned}$$

By defining $\mathbf{O} = [o_{ij}] \in \mathbb{R}_+^{N\times N}$, we rewrite (13) as

$$\begin{aligned}\mathcal{R}_{\text{LSE}}(\mathbf{O}, \mathbf{H}) &= \sum_{i=1}^{N}\mathbf{h}_i^T\mathbf{h}_i d_{ii} - \sum_{i\neq j}\mathbf{h}_i^T\mathbf{h}_j o_{ij} \\ &= \text{Tr}\left(\mathbf{H}^T\mathbf{D}\mathbf{H}\right) - \text{Tr}\left(\mathbf{H}^T\mathbf{O}\mathbf{H}\right) = \text{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right) \quad (14)\end{aligned}$$

where Tr$(\cdot)$ denotes the trace of a matrix, $\mathbf{D} = [d_{ij}] \in \mathbb{R}_+^{N\times N}$ is a diagonal matrix whose entries are row summation of $\mathbf{O}$, i.e., $d_{ii} = \sum_{j=1}^{N} o_{ij}$, and $\mathbf{L} = \mathbf{D} - \mathbf{O}$ is the graph regularization matrix (Laplacian matrix) of prior information $\mathbf{O}$. Similarly, the KL-divergence based constraints can be written as

$$\mathcal{R}_{\text{KL}}(\mathbf{O}, \mathbf{H}) = \frac{1}{2}\sum_{(i,j,o_{ij})\in C} o_{ij}\left(\mathcal{D}_{\text{KL}}\left(\mathbf{h}_i || \mathbf{h}_j\right) + \mathcal{D}_{\text{KL}}\left(\mathbf{h}_j || \mathbf{h}_i\right)\right) \qquad (15)$$

which takes into account the asymmetry of KL-divergence and averages $\mathcal{D}_{\text{KL}}(\mathbf{h}_i||\mathbf{h}_j)$ and $\mathcal{D}_{\text{KL}}(\mathbf{h}_j||\mathbf{h}_i)$. By minimizing $\mathcal{R}_{\text{LSE}}(\mathbf{O}, \mathbf{H})$ or $\mathcal{R}_{\text{KL}}(\mathbf{O}, \mathbf{H})$, we expect the new representations of two nodes i and j are similar if we have some information indicating they might belong to the same community, i.e., the corresponding element o_{ij} is not zero.

Now that we have the graph regularization term $\mathcal{R}_\beta(\mathbf{O}, \mathbf{H}), \beta \in \{\text{LSE}, \text{KL}\}$, we incorporate them into foundational object function of topology information as

$$\mathcal{F}_{\alpha,\beta}(\mathbf{H}|\mathbf{A}, \mathbf{O}) = \mathcal{L}_\alpha(\mathbf{A}, \mathbf{H}) + \lambda\mathcal{R}_\beta(\mathbf{O}, \mathbf{H}) \qquad (16)$$

where $\alpha \in \{\text{LSE}, \text{KL}, \text{SYM}, \text{MOD}, \text{ADJ}, \text{LAP}, \text{NLAP}\}$ as show in Section III and λ is the parameter to balance the tradeoff between topology information and prior information. In most cases, the sign of λ is positive. Only when the first term is $\mathcal{L}_{\text{ADJ}}$ or $\mathcal{L}_{\text{MOD}}$, the sign of λ is negative since these term need be maximized while the second term need be minimized. For computational simplicity, we choose the same distance function for two parts in (16). $\beta = \text{LSE}$ when $\alpha \in \{\text{LSE}, \text{SYM}, \text{MOD}, \text{ADJ}, \text{LAP}, \text{NLAP}\}$, and $\beta = \text{KL}$ when $\alpha = \text{KL}$. Thus we obtain the following objective functions:

$$\mathcal{F}_{\text{LSE}}(\mathbf{H}|\mathbf{A}, \mathbf{O}) = \left|\left|\mathbf{A} - \mathbf{W}\mathbf{H}^T\right|\right|_F^2 + \lambda Tr\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right) \qquad (17)$$

$$\mathcal{F}_{\text{SYM}}(\mathbf{H}|\mathbf{A}, \mathbf{O}) = \left|\left|\mathbf{A} - \mathbf{H}\mathbf{H}^T\right|\right|_F^2 + \lambda Tr\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right) \qquad (18)$$

$$\begin{aligned}\mathcal{F}_{\text{KL}}(\mathbf{H}|\mathbf{A}, \mathbf{O}) = &\sum_{i=1}^{N}\sum_{j=1}^{N}\Bigg(a_{ij}\log\left(\frac{a_{ij}}{\sum_{k=1}^{K} w_{ik}h_{jk}}\right) \\ &\qquad - a_{ij} + \sum_{k=1}^{K} w_{ik}h_{jk}\Bigg) \\ &+ \frac{\lambda}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K}\Bigg(h_{ik}\log\left(\frac{h_{ik}}{h_{jk}}\right) \\ &\qquad + h_{jk}\log\left(\frac{h_{jk}}{h_{ik}}\right)\Bigg) o_{ij} \qquad (19)\end{aligned}$$

$$\mathcal{F}_{\text{MOD}}(\mathbf{H}|\mathbf{A}, \mathbf{O}) = -\,\text{Tr}\left(\mathbf{H}^T\mathbf{B}\mathbf{H}\right) + \lambda\,\text{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right) \qquad (20)$$

$$\mathcal{F}_{\text{LAP}}(\mathbf{H}|\mathbf{A}, \mathbf{O}) = \text{Tr}\left(\mathbf{H}^T(\mathbf{D} - \mathbf{A})\mathbf{H}\right) + \lambda\,\text{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right). \qquad (21)$$

Semi-supervised SC based on adjacency matrix $\mathcal{F}_{\text{ADJ}}$ and normalized Laplacian matrix $\mathcal{F}_{\text{NLAP}}$ are similar to $\mathcal{F}_{\text{MOD}}$ and

$\mathcal{F}_{\text{LAP}}$ except that we take $\mathbf{A}$ and $\mathbf{I}-\mathbf{D}^{-1}\mathbf{A}$ instead of $\mathbf{B}$ and $\mathbf{D}-\mathbf{A}$. Details on how to solve these optimization problems are presented in the next section.

B. Algorithms

1) Algorithms for Semi-Supervised SC: Since all the four semi-supervised SC algorithms $\mathcal{F}_\alpha(\mathbf{H}|\mathbf{A},\mathbf{O}), \alpha \in \{\text{MOD}, \text{ADJ}, \text{LAP}, \text{NLAP}\}$ can be written as a uniform formulation as

$$\begin{aligned}\mathcal{F}_{SC}(\mathbf{H}|\mathbf{A},\mathbf{O}) &= \operatorname{Tr}\left(\mathbf{H}^T\mathbf{G}\mathbf{H}\right) + \lambda \operatorname{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right)\\ &= \operatorname{Tr}\left(\mathbf{H}^T(\mathbf{G}+\lambda'\mathbf{L})\mathbf{H}\right) \qquad (22)\end{aligned}$$

where $\lambda' = \lambda/N$ and $\mathbf{G}$ equals to $-\mathbf{B}$, $-\mathbf{A}$, $\mathbf{D}-\mathbf{A}$ and $\mathbf{I}-\mathbf{D}^{-1}\mathbf{A}$ when $\alpha = \text{MOD}, \text{ADJ}, \text{LAP}, \text{NLAP}$, respectively. It can be optimized by finding the eigenvectors corresponding to the smallest eigenvalues as in (10).

2) Algorithms for Semi-Supervised NMF: Since the objective functions $\mathcal{F}_{\text{LSE}}$, $\mathcal{F}_{\text{SYM}}$, and $\mathcal{F}_{\text{KL}}$ of our framework in (17)–(19) are not convex in both $\mathbf{H}$ and $\mathbf{W}$ as in the original NMF model, it is highly unlikely to develop an algorithm to find the global minima. In this section, we develop three iterative algorithms as in [40], which can achieve local minima, for these objective functions.

Use $\mathcal{F}_{\text{LSE}}$ in (17) as an example, we introduce how to minimize the objective function. Using some properties of the trace and Frobenius norm of square matrix, i.e., $\text{Tr}(\mathbf{A}) = \text{Tr}(\mathbf{A}^T)$, $\text{Tr}(\mathbf{AB}) = \text{Tr}(\mathbf{BA})$ and $||\mathbf{A}||_F^2 = \text{Tr}(\mathbf{AA}^T)$, we can rewrite $\mathcal{F}_{\text{LSE}}$ as

$$\begin{aligned}&\mathcal{F}_{\text{LSE}}(\mathbf{H}|\mathbf{A},\mathbf{O})\\ &\quad= \operatorname{Tr}\left(\left(\mathbf{A}-\mathbf{W}\mathbf{H}^T\right)\left(\mathbf{A}-\mathbf{W}\mathbf{H}^T\right)^T\right) + \lambda \operatorname{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right)\\ &\quad= \operatorname{Tr}\left(\mathbf{A}\mathbf{A}^T\right) + Tr\left(\mathbf{W}\mathbf{H}^T\mathbf{H}\mathbf{W}^T\right) - 2\operatorname{Tr}\left(\mathbf{A}\mathbf{H}\mathbf{W}^T\right)\\ &\quad+ \lambda \operatorname{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right). \qquad (23)\end{aligned}$$

By introducing Lagrange multiplier $\Psi = [\psi_{ij}] \in \mathbb{R}^{N\times K}$ and $\Phi = [\phi_{ij}] \in \mathbb{R}^{N\times K}$ for constraints $\mathbf{W} = [w_{ij}] \geq 0$ and $\mathbf{H} = [h_{ij}] \geq 0$, respectively, we write the Lagrange $\mathcal{L}_{\text{LSE}}$ as

$$\begin{aligned}\mathcal{L}_{\text{LSE}} &= \operatorname{Tr}\left(\mathbf{A}\mathbf{A}^T\right) + \operatorname{Tr}\left(\mathbf{W}\mathbf{H}^T\mathbf{H}\mathbf{W}^T\right) - 2\operatorname{Tr}\left(\mathbf{A}\mathbf{H}\mathbf{W}^T\right)\\ &\quad+ \lambda \operatorname{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right) + \operatorname{Tr}\left(\Psi\mathbf{W}^T\right) + \operatorname{Tr}\left(\Phi\mathbf{H}^T\right). \qquad (24)\end{aligned}$$

To find $\mathbf{W}$ and $\mathbf{H}$ which minimize $\mathcal{L}_{\text{LSE}}$, we iteratively minimize one matrix variable while fixing another. Because $\mathcal{L}_{\text{LSE}}$ is differentiable with respect to $\mathbf{W}$ and $\mathbf{H}$, we obtain the partial derivatives of $\mathcal{L}_{\text{LSE}}$ with respect to $\mathbf{W}$ and $\mathbf{H}$ as

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{LSE}}}{\partial \mathbf{W}} &= -2\mathbf{A}\mathbf{H} + 2\mathbf{W}\mathbf{H}^T\mathbf{H} + \Psi\\ \frac{\partial \mathcal{L}_{\text{LSE}}}{\partial \mathbf{H}} &= -2\mathbf{A}^T\mathbf{W} + 2\mathbf{H}\mathbf{W}^T\mathbf{W} + 2\lambda\mathbf{L}\mathbf{H} + \Phi. \qquad (25)\end{aligned}$$

Combining the equations that partial derivatives equal to zero and the Karush–Kuhn–Tucker conditions $\psi_{ik}w_{ik} = 0$ and $\phi_{jk}h_{jk} = 0$, we obtain the solutions of w_{ik} and h_{jk}

$$\begin{aligned}-(\mathbf{A}\mathbf{H})_{ik}w_{ik} + \left(\mathbf{W}\mathbf{H}^T\mathbf{H}\right)_{ik}w_{ik} &= 0\\ -\left(\mathbf{A}^T\mathbf{W}\right)_{jk}h_{jk} + \left(\mathbf{H}\mathbf{W}^T\mathbf{W}\right)_{jk}h_{jk} + \lambda(\mathbf{L}\mathbf{H})_{jk}h_{jk} &= 0. \qquad (26)\end{aligned}$$

Finally, we obtain the following updating rules:

$$w_{ik} \leftarrow w_{ik}\frac{(\mathbf{A}\mathbf{H})_{ik}}{\left(\mathbf{W}\mathbf{H}^T\mathbf{H}\right)_{ik}}, \; h_{jk} \leftarrow h_{jk}\frac{\left(\mathbf{A}^T\mathbf{W} + \lambda\mathbf{O}\mathbf{H}\right)_{jk}}{\left(\mathbf{H}\mathbf{W}^T\mathbf{W} + \lambda\mathbf{D}\mathbf{H}\right)_{jk}}. \qquad (27)$$

When λ equals to zero, the updating rules in (27) reduce to (5) which is the updating rule of standard NMF based on Euclidean distance.

Similarly, we obtain the updating rule to minimize $\mathcal{F}_{\text{SYM}}$ in (18) as

$$h_{ik} \leftarrow h_{ik}\frac{(\mathbf{A}\mathbf{H} + \lambda''\mathbf{O}\mathbf{H})_{ik}}{\left(\mathbf{H}\mathbf{H}^T\mathbf{H} + \lambda'\mathbf{D}\mathbf{H}\right)_{ik}} \qquad (28)$$

in which we set $\lambda'' = 2\lambda$ for the consistency of the formulas.

Finally, we introduce the updating rules to minimize $\mathcal{F}_{\text{KL}}$ in (19) as

$$\begin{aligned}w_{ik} &\leftarrow w_{ik}\frac{\sum_j \left(a_{ij}h_{jk}/\sum_k w_{ik}h_{jk}\right)}{\sum_j h_{jk}}\\ \mathbf{h}_k &\leftarrow \left(\sum_i w_{ik}\mathbf{I} + \lambda\mathbf{L}\right)^{-1}\hat{\mathbf{h}}_k \qquad (29)\end{aligned}$$

in which $\mathbf{h}_k$ is the kth column of $\mathbf{H}$, $\mathbf{I}$ is an $N\times N$ identity matrix and

$$\hat{\mathbf{h}}_k = \begin{bmatrix} h_{1k}\sum_i \left(a_{i1}w_{ik}/\sum_k w_{ik}h_{1k}\right)\\ h_{2k}\sum_i \left(a_{i2}w_{ik}/\sum_k w_{ik}h_{2k}\right)\\ \vdots\\ h_{Nk}\sum_i \left(a_{iN}w_{ik}/\sum_k w_{ik}h_{Nk}\right)\end{bmatrix}. \qquad (30)$$

It is proved that the updating rules in (27) and (29) can find the local minima of objective functions $\mathcal{F}_{\text{LSE}}(\mathbf{H}|\mathbf{A},\mathbf{O})$ in (17) and $\mathcal{F}_{\text{KL}}(\mathbf{H}|\mathbf{A},\mathbf{O})$ in (19), respectively [40].

C. Complexity Analysis

1) Complexity Analysis for Semi-Supervised SC: As mentioned above, the algorithm for solving semi-supervised spectral problem is the same as that for common SC, i.e., eigenvalue decomposition. In theory, it has the same complexity of matrix multiplication whose upper bound is $O(N^3)$ where N is the number of nodes. In practice, its computational complexity is $O(N^{2.376})$ using Coppersmith and Winograd's [41] algorithm. Besides, due to the sparsity and symmetry of the adjacency matrix $\mathbf{A}$, we can make use of some existing softwares, such as ARnoldi PACKage, to solve the large-scale eigenvalue problem.

2) Complexity Analysis for Semi-Supervised NMF: In this section, we analyze the computational complexity of our proposed semi-supervised framework based on two different vector distance metrics, i.e., Frobenius norm based methods [$\mathcal{F}_{\text{LSE}}$ in (17) and $\mathcal{F}_{\text{SYM}}$ in (18)] and KL-divergence based method [$\mathcal{F}_{\text{KL}}$ in (19)]

For algorithms based on Frobenius norm illustrated in (27) and (28), each iteration in the updating process needs $O(N^2K)$ floating point operations by considering the number of communities $K \ll N$. Therefore, our framework does not increase the complexity of standard NMF algorithms. By taking into account the sparsity of the adjacency matrix $\mathbf{A}$, the complexity of each iteration is reduced to $O((NK+M+R)K)$, in which R is the number of must-link priors. If the number of the added must-links R is in the same order of magnitude with the number of edges M, the proposed framework has the same computational complexity with the original NMF methods.

In (29), the updating rules for KL-divergence based formulation need to calculate the inverse of a matrix of size $N \times N$, whose computational complexity is $O(N^3)$. In practice, however, it only requires to solve the linear equations system $\mathbf{Ax} = \mathbf{b}$. We make use of conjugate gradient method [42] to solve this linear equations system, because $\sum_i w_{ik}\mathbf{I} + \lambda\mathbf{L}$ is a symmetric positive definite matrix. Since the percentage of prior information we add to the framework is small, the matrix to be inversed is sparse. Suppose there are average p nonzero elements in each column of $\sum_i w_{ik}\mathbf{I} + \lambda\mathbf{L}$, and conjugate gradient method needs q iterations to converge, we need $O(q(p+4)N)$ float point operations to obtain the solution of the linear equations system as shown in [40]. In general, conjugate gradient algorithms can converge in few iterations. And in our experiments, we set the maximum iterations to 20. Usually, conjugate gradient only needs about 20 iterations to coverage. Besides, as there are K linear equations systems to be solved, the overall computational cost for each iteration is

$$O\left(N^2K + q(p+4)NK\right) = O\left((N + q(p+4))NK\right).$$

From the above complexity analysis, we note that Frobenius norm based algorithms, i.e., $\mathcal{F}_{\mathrm{LSE}}$ and $\mathcal{F}_{\mathrm{SYM}}$, are more suitable for large scale networks, since their complexities are near linear with network size N. Besides, we can make use of parallel [43] and distributed [44] computing to make our framework applicable to more large-scale networks, since there are many parallel algorithms on eigenvalue decomposition, matrix multiplication and NMF have been proposed.

D. Model Selection

Model selection, which is an important problem in community detection, is to determine the number of communities K in the networks. There are several model selection strategies available, such as consensus clustering, eigenvalue gaps [29], [45], cross-validation and Bayes Information Criterion [46]. However, in order to make our framework more uniform, we adopt the widely-used modularity Q [shown in (9)] as the criterion to determine the number of communities K, as done in [6], [22], and [23]. The great advantage of this scheme is that it is independent of the specific community detection algorithms. On networks which we do not know the number of communities, we can choose K which corresponds to the maximal modularity Q.

V. Experiments

To test the performance of the proposed semi-supervised community detection framework, we verify the performance improvement both on two artificial network benchmarks and on some widely used real-world networks as shown in Table I, and compare it with a state-of-the-art method. To illustrate the broad applicability of the framework, we apply it to all the algorithms mentioned in the Section IV, i.e., square distance based NMF (NMF_LSE), KL-divergence based NMF (NMF_KL), SNMF, adjacency matrix based SC (SC_ADJ), standard Laplacian matrix based SC (SC_LAP) and normalized Laplacian matrix based SC (SC_NLAP).

TABLE I
Real-World Networks

Datasets	N	M	K	Description
Karate [47]	34	78	2	Zachary's karate club
Dolphins [48]	62	159	2	Dolphin social network
Friendship6 [49]	68	220	6	High school friendship
Friendship7 [49]	68	220	7	High school friendship
Word [29]	112	425	2	Word network
Football [1]	115	613	12	American College football
Polbooks [6]	105	441	3	Books about US politics
Polblogs [12]	1,490	16,718	2	Blogs about US politics

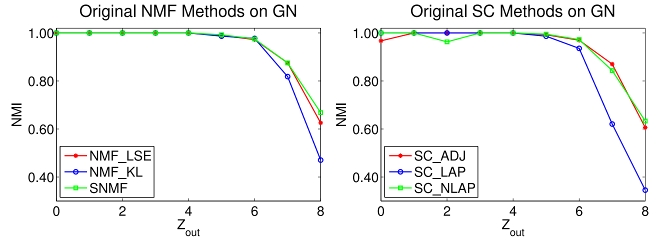


Fig. 3. Performance of original methods as a function of the number of intercommunity edges per vertex, Z_{out}, on GN networks. As Z_{out} increases, all the methods degrade and even fail when $Z_{\mathrm{out}} \geq 7$.

All experiments are conducted on a single PC (Intel Core i7-2600 CPU @ 3.40 GHz. Processor with 4 G memory). The source code of all the algorithms used in this paper can be downloaded from authors' websites. There are totally $N(N-1)/2$ pairs of membership in a undirected network with N nodes, while the number of pairs that indicate the two nodes belong to the same community are

$$N_{\mathrm{pairs}} = \sum_{k=1}^{K} N_k(N_k - 1)/2 \tag{31}$$

in which K is the number of communities and N_k is the number of nodes in the kth ground-truth community. The percentage we used in this paper is based on the N_{pairs}.

We use normalized mutual information (NMI) [50] to evaluate the performance of the community detection. NMI is more informative than just simply counting the number of misclassified nodes. It especially suitable for imbalanced datasets such as Lancichinetti–Fortunato–Radicchi networks (LFR networks) benchmark and some real-world networks which will be discussed in the following sections.

A. Artificial Benchmark Networks

The Girvan–Newman networks (GN networks) benchmark [1] is a type of basic benchmark networks for testing community detection algorithms. Each network consists of 128 vertices which are divided into four communities of 32 vertices each. Each vertex has on average 16 edges which randomly connect to Z_{in} vertices in the own community and Z_{out} vertices in other communities, and $Z_{\mathrm{in}} + Z_{\mathrm{out}} = 16$. For each pair of Z_{in} and Z_{out}, we randomly generate ten networks. Obviously, the community structure is clear when Z_{out} is

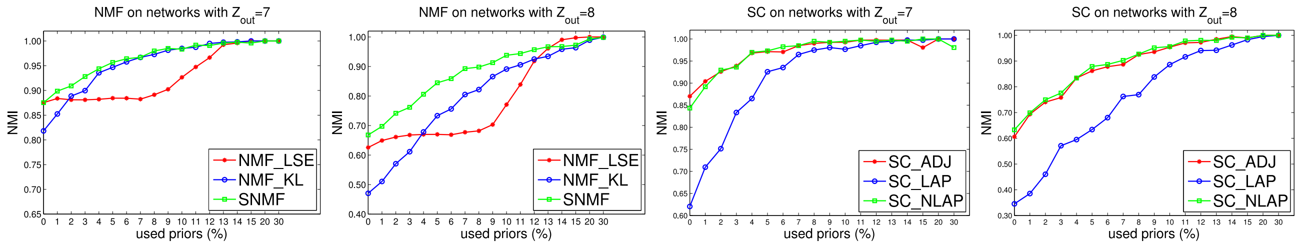


Fig. 4. Performance of our framework in terms of NMI as a function of the percentage of priors added on GN networks. The left two plots are based on NMF methods, while the right two are based on SC methods.

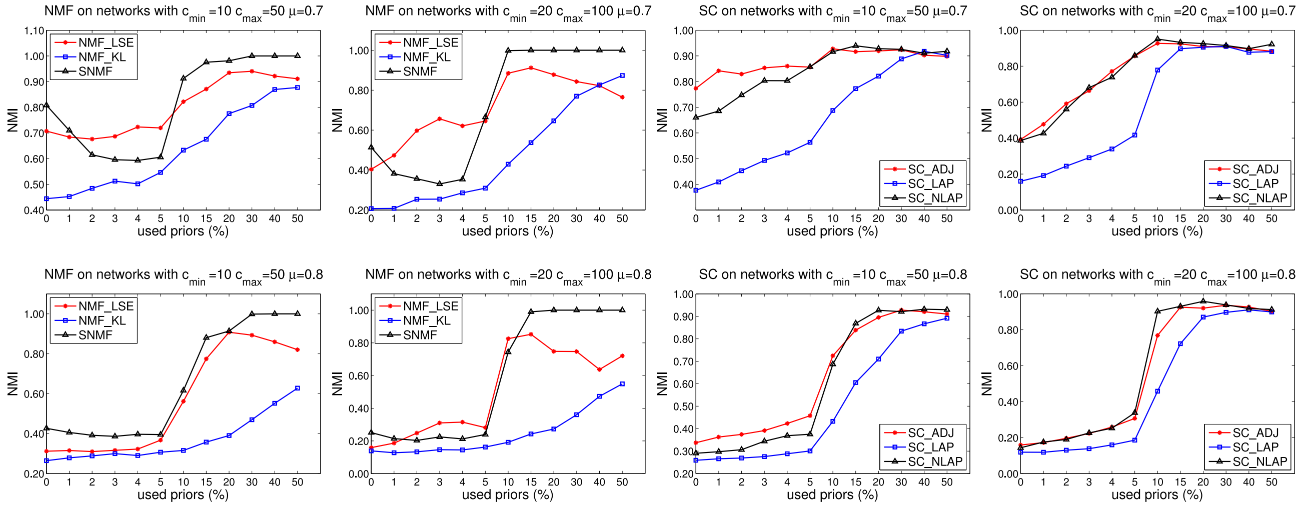


Fig. 5. Performance of our framework in terms of NMI as a function of the percentage of priors added on LFR networks. The first two columns are based on NMF methods, while the last two columns are based on SC methods. We tune the generator parameters to make networks prefer small community size, i.e., 10–50, (in odd columns) or large community size, i.e., 20–100, (in even columns). We also vary mixing parameter (μ) values to show how our framework performs on clear ($\mu = 0.7$, top row) and vague ($\mu = 0.8$, bottom row) networks.

small. And, as the Z_{out} increases, the structure of the network becomes vague, and the task becomes challenging. In Fig. 3, we plot the NMI of original NMF and SC methods as a function of Z_{out}. It is easy to find that all the methods achieve good performance in the networks with $Z_{\text{out}} \leq 6$. And as $Z_{\text{out}} > 6$, all the methods degrade significantly. Especially, the worst performer reaches 0.35 when $Z_{\text{out}} = 8$. It implies that the topology information becomes insufficient to accurately discover the community when $Z_{\text{out}} > 6$ and needs the help of the prior information. So we focus on the networks whose Z_{out} equals to 7 or 8 in the experiments on GN networks.

To validate our framework, we first fix the tradeoff parameter λ to 1 and set $o_{ij} = 1$ when we have the prior that nodes i and j belong to the same community. We display the average performance of our framework based on different methods in Fig. 4. The NMIs of all the methods increase consistently as the used priors. All of them reach 1 when prior information is adequate. This validate the effectiveness of our framework. Besides, various methods have different growth trends. Nevertheless, the beneficial gained from the same percentage of priors are more obvious on vague networks (e.g., $Z_{\text{out}} = 8$) than on clear ones (e.g., $Z_{\text{out}} = 7$). It meets the motivation of our framework that by encoding prior information we enhance the performance of community detection in networks which do not have clear structure.

Though GN networks benchmark is a popular benchmark for community detection, the community structures are much more complex in real life: the network is large, the number of vertices in different communities are distinct and there is great difference between nodes' degree. The LFR networks benchmark [51] aims at addressing the above problems. LFR generator allows to specify the number of nodes (N), average degree (k), community size distribution (β), degree distribution (γ), minimum and maximum of the community sizes ($c_{\min}$ and $c_{\max}$), and the fraction of intercommunity edge (mixing parameter μ). In LFR, both community size and degree distributions are power laws, from which vertices and communities are generated by sampling. Similar to the role of Z_{out} in the GN benchmark, μ in LFR networks benchmark controls the clarity of the network structure. With the increase of μ, the structure of network becomes vague, and the detection of communities becomes more difficult.

In this paper, we follow experiment setting designed by Lancichinetti *et al.* [51], and set the number of nodes to 1000, the minimum community size to 10 or 20, the maximum community size to five times the minimum community size, average degree as 20, the exponent of the vertex degree and community size as -2 and -1, respectively, and mixing parameter as several different values, 0.7 and 0.8. We also fix the tradeoff parameter λ as in the experiments of GN networks. In Fig. 5, we show the average results of our

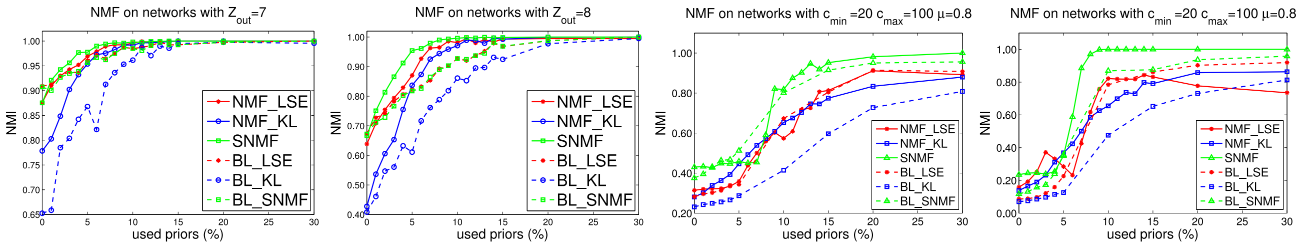


Fig. 6. Performance comparison with other semi-supervised approaches. BL_LSE, BL_KL, and BL_SNMF are the methods under the framework of [22] which directly connects the must-links. The left two plots are the results on GN networks, while the right two are that on LFR networks.

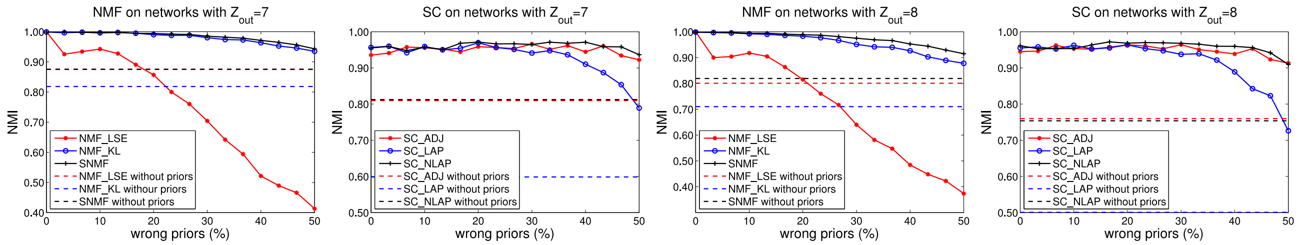


Fig. 7. Impact of wrong priors on performance. The horizontal dashed lines are the results from methods without priors. By adding 30% of correct priors, all methods achieve satisfactory results. We plot the impact of wrong priors on performance by varying the wrong priors percentage from 0% to 50%.

framework on LFR networks. It is easy to find that the role of our framework is more significant in networks with unclear structures, i.e., large mixing parameter μ networks. The three different SC methods have similar performance, i.e., the NMIs increase consistently with the increase of used priors. The three different NMF-based methods have their own strengths and weaknesses. Though the basic NMF_KL has the lowest NMI without prior, it continues to increase as the percentage of prior increases. NMF_LSE has better original performance than NMF_KL, but it degrades after the percentage achieves 30%. Although the original performance of SNMF is the best of all the three methods and it can achieve 1 rapidly after prior percent exceeds 4% or 5%, it may degrade when the percentage of prior is small, especially when the μ is small, which means the community structure is clear. The reason why the performance of SNMF degrades may be that the adjacency matrix is factorized into the product of one matrix and its transposition which makes the priors cannot propagate to other part of the network. Besides, the randomly selected priors also affect the topology of the original networks. Especially, in networks with a large number of communities, the randomly selected priors often make some parts of one community form some small communities instead of a big one. From this experiment, we find that the framework based on SNMF is suitable for the situation with sufficient prior information, while that based on NMF_LSE is suitable for limited priors.

To illustrate the performance of our framework, we compare it with the framework proposed by Zhang [22], which directly modifies the network topology by connecting the must-link constraints. Since both these two frameworks result in the same optimization formulation on semi-supervised SC approaches, we only compare their performance on NMF approaches. On all the networks, we set the parameter λ to 10 for NMF_KL and SNMF. In Fig. 6, we display the performance on GN networks (the left two figures) and LFR networks (the right two figure) with different network settings. We find most of our methods significantly outperform the corresponding methods under Zhang's framework (BL_LSE, BL_KL, and BL_SNMF) except for NMF_LSE on LFR networks. This further implies the efficiency of our prior information encoding strategy.

B. Wrong Priors Impact

In general, priors are considered as the correct labels from human, but wrong labels are also unavoidable in practice. To demonstrate the robustness of our framework, we investigate the impact of wrong priors on performance. As we can see from Fig. 4, most methods can achieve satisfactory results with 30% correct priors. Thus we add 30% of priors, part of which are not correct, to the framework, and vary the wrong priors percentage from 0% to 50%. In Fig. 7, we plot the impact of wrong priors on performance. By introducing 50% wrong priors, i.e., 15% correct priors and 15% wrong priors, most methods (NMF_KL, SNMF, SC_LAP, and SC_NLAP) only decrease about 0.1, and the results are still higher than original methods without priors. This shows that our framework are robust to noises and wrong priors. The reason why NMF_LSE is sensitive to wrong priors may be that we only impose priors constraints on one factorized matrix.

C. Parameter Setting

To illustrate the effect of the tradeoff parameter and discuss how to determine it, we evaluate the role of balancing parameter λ in (16). In Fig. 8, we plot the performance of our algorithms as the λ varies from 0.1 to 10. The results come from six algorithms on two networks with Z_{out} equals 7 (odd rows) and 8 (even rows). For better illustration, we only select a small portion of λ values. Because our added priors are accurate ones, the performance of our algorithms consistently increase when we weight more on prior information. Taking SNMF on networks with $Z_{\text{out}} = 8$ as an example, as

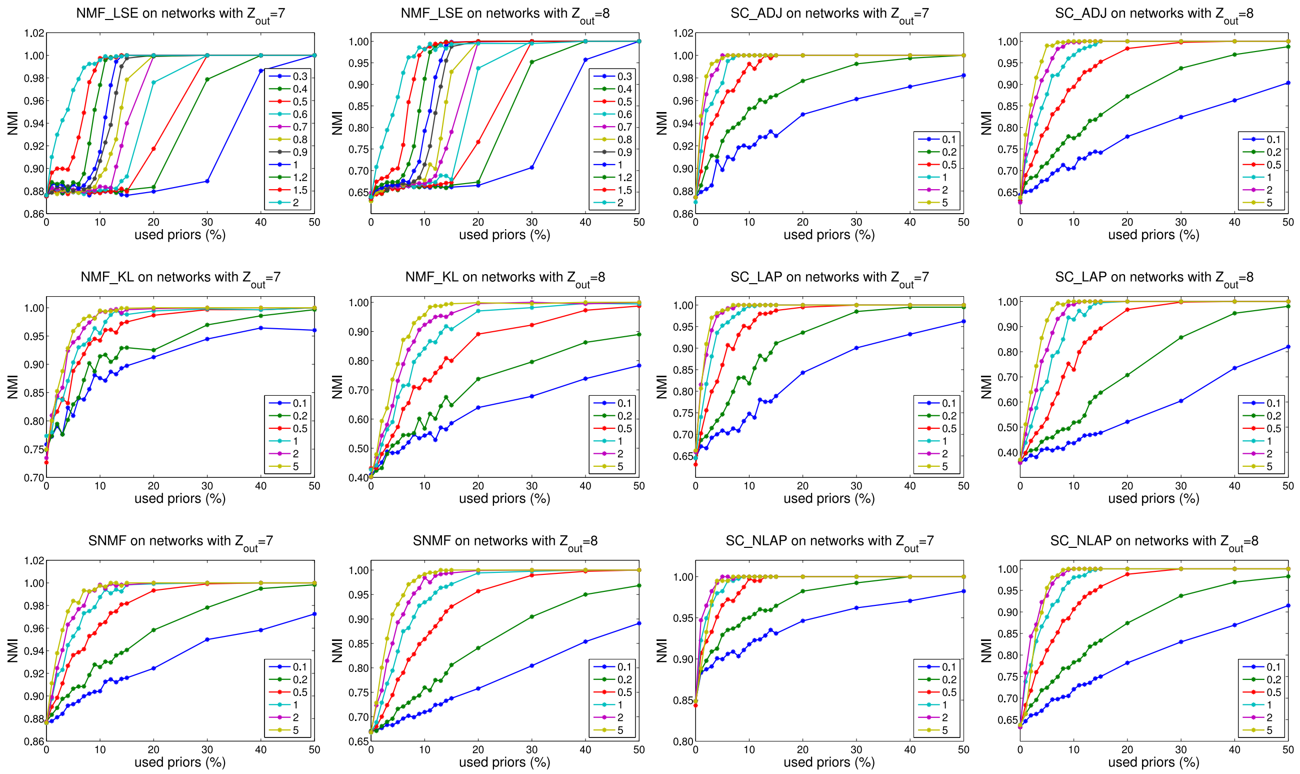


Fig. 8. Performance of our framework with respect to the balancing parameter λ on GN networks. The left two columns are the results of our framework based on NMF, while the right two are the results of our framework based on SC.

15% priors used, the NMI reaches 0.73, 0.81, 0.93, 0.97, 0.99, and 1 when we set λ as 0.1, 0.2, 0.5, 1, 2, and 5, respectively. Furthermore, to make NMI achieve 1, we need to add 13%, 30%, 40%, and 50% priors when λ equals 5, 2, 1, and 0.5.

Besides, we show the effect of the balancing parameter λ on LFR benchmarks in Figs. 9 and 10. We first display the performances of all the methods based on different λ values on the network whose community scale is between 10 and 50 and mixture parameter $\mu = 0.8$ to show the overall impact on it in Fig. 9. And then we give a detailed analysis of the effect of λ in SNMF on networks with different mixing parameter μ values in Fig. 10. The reason why we choose SNMF as an example is that SNMF-based framework is the most sensitive one among all the methods shown in Fig. 5. The settings of the networks in Fig. 10 are the same as the second column in Fig. 5. Curves in each figure are based on the different tradeoff parameters λ values varying from 0.1 to 5. From these curves we obtain the following findings and conclusions.

1) As shown in Fig. 9, the impacts of λ in most of the methods are positive. In other word, with the increase of λ, most of these methods can achieve a much higher performance with the same priors.
2) If the network structure is unclear, e.g., the right figure in Fig. 10, increasing λ can significantly and consistently improve the performance. This also means the prior information plays a more important role in detecting on complicated networks. Thus we can appropriately choose a large parameter λ to highlight this effect on this kind of networks.
3) In the networks with clear structure, such as the left figure in Fig. 10, the performance of SNMF is sensitive to the parameter λ. But this sensitivity will decrease with the decrease of the clarity of the networks topology as shown in the right figure in Fig. 10.
4) In the left figure of Fig. 10, the performance may degrade when we use a large λ with limited prior information, and this degradation becomes serious as λ increases. Thus we should choose a relatively small λ in SNMF when the prior information is limited.

In summary, the equal contribution of topology structure and prior information, i.e., $\lambda = 1$, mostly achieves satisfactory results. If we have some prior that the structure of network is not very clear, we can increase λ appropriately, and vice versa.

D. Real-World Networks

In this section, we evaluate our framework on eight widely used real networks which are shown in Table I. Here N, M, and K denote the number of vertices, edges, and communities, respectively. To save space we only select four different methods NMF_LSE, NMF_KL, SC_ADJ, and SC_LAP to demonstrate their improvements. The results on real-world networks are shown in Table II. From the results we can find that the NMIs of all the methods increase significantly with the increase of used priors although there exists some local nonsmoothness. On clear structure networks where original methods can achieve good performance, e.g., Zacharys karate club network and American college football network,

TABLE II
PERFORMANCE OF OUR FRAMEWORK BASED ON FOUR DIFFERENT METHODS ON EIGHT REAL-WORLD NETWORKS

Prior	**Word**				**Dolphins**				**Football**				**Karate**			
Percent	**LSE**	**KL**	**ADJ**	**LAP**	**LSE**	**KL**	**ADJ**	**LAP**	**LSE**	**KL**	**ADJ**	**LAP**	**LSE**	**KL**	**ADJ**	**LAP**
Baseline	0.341	0.418	0.013	0.001	0.814	0.864	0.142	0.889	0.927	0.897	0.919	0.922	1.000	1.000	1.000	0.836
2%	0.341	0.583	0.009	0.022	0.814	0.938	0.692	0.978	0.926	0.909	0.927	0.921	1.000	1.000	1.000	0.935
5%	0.352	0.740	0.009	0.023	0.814	1.000	0.918	1.000	0.927	0.913	0.920	0.925	1.000	1.000	1.000	0.935
8%	0.807	0.906	0.016	0.040	0.814	1.000	0.956	1.000	0.924	0.920	0.930	0.920	1.000	1.000	1.000	0.935
10%	0.914	0.957	0.021	0.021	0.814	1.000	0.956	0.978	0.925	0.911	0.923	0.924	1.000	1.000	1.000	1.000
15%	1.000	1.000	0.133	0.392	1.000	1.000	1.000	1.000	0.926	0.928	0.931	0.918	1.000	1.000	1.000	1.000
20%	1.000	1.000	0.572	0.730	1.000	1.000	1.000	1.000	0.925	0.921	0.920	0.925	1.000	1.000	1.000	1.000
30%	1.000	1.000	0.961	0.987	1.000	1.000	1.000	1.000	0.929	0.943	0.943	0.942	1.000	1.000	1.000	1.000
Prior	**Polblogs**				**Polbooks**				**Friendship6**				**Friendship7**			
Percent	**LSE**	**KL**	**ADJ**	**LAP**	**LSE**	**KL**	**ADJ**	**LAP**	**LSE**	**KL**	**ADJ**	**LAP**	**LSE**	**KL**	**ADJ**	**LAP**
Baseline	0.527	0.527	0.014	0.001	0.540	0.562	0.505	0.574	0.798	0.874	0.786	0.892	0.836	0.899	0.802	0.851
2%	1.000	0.965	0.310	0.966	0.532	0.614	0.388	0.616	0.798	0.909	0.861	0.894	0.843	0.908	0.825	0.874
5%	1.000	0.990	0.611	0.986	0.578	0.664	0.570	0.660	0.798	0.936	0.907	0.930	0.843	0.912	0.857	0.885
8%	1.000	0.995	0.974	0.993	0.676	0.763	0.626	0.710	0.801	0.909	0.887	0.950	0.833	0.921	0.898	0.934
10%	1.000	1.000	0.987	1.000	0.698	0.759	0.660	0.722	0.803	0.952	0.917	0.940	0.843	0.951	0.888	0.925
15%	1.000	1.000	1.000	1.000	0.787	0.774	0.717	0.769	0.808	0.937	0.927	0.960	0.846	0.930	0.913	0.935
20%	1.000	1.000	1.000	1.000	1.000	0.866	0.810	0.843	0.824	0.963	0.929	0.962	0.846	0.977	0.920	0.954
30%	1.000	1.000	1.000	1.000	1.000	0.946	0.925	0.937	0.889	0.967	0.951	0.965	0.884	0.985	0.943	0.973

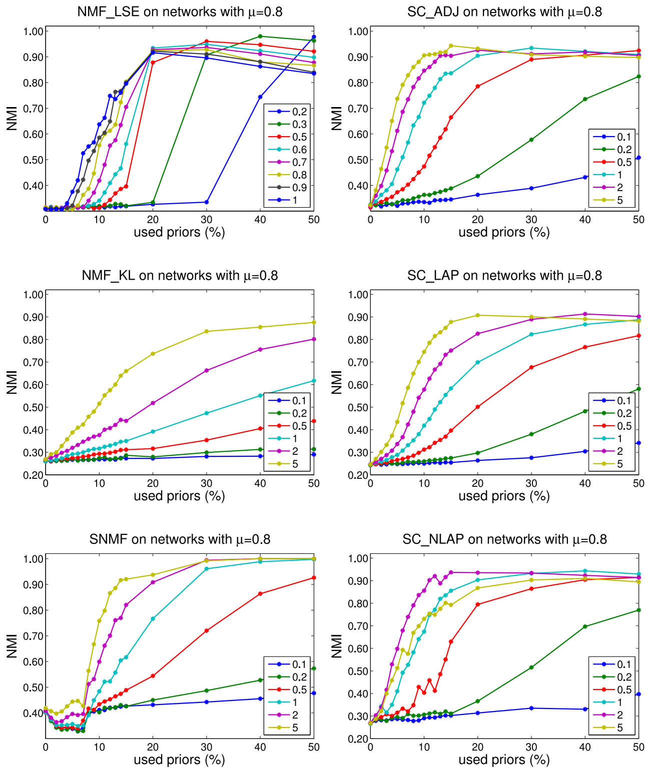


Fig. 9. Performance of our framework with respect to the balancing parameter λ on LFR networks whose mixing parameters μ are fixed. The first column is based on NMF, while the second column is based on SC. Each curve is the result of one method under a certain λ value.

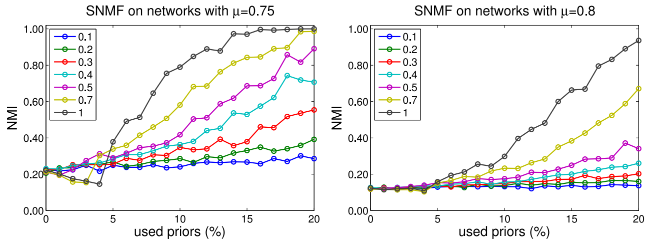


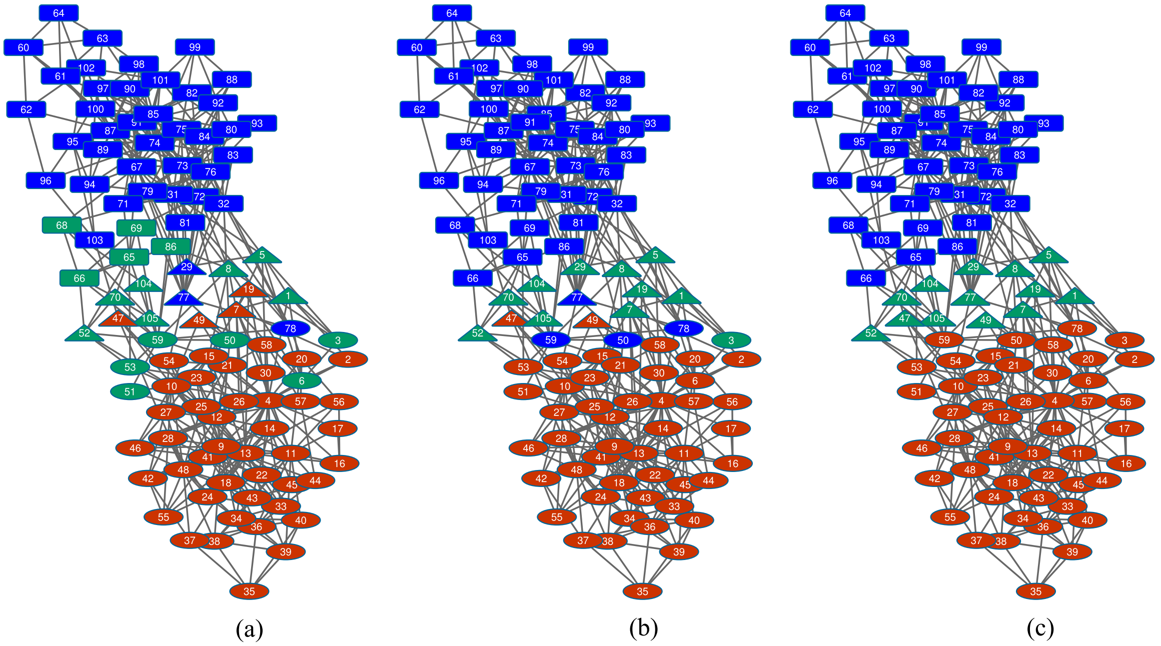
Fig. 10. Performance of our framework based on the SNMF with respect to the balancing parameter λ and mixing parameter (μ) values on LFR networks. Curves in each figure are based on the different λ.

our framework also do not degrade. There are 15 methods on networks reach 1 as we integrate 30% of prior information, while there are only three of them achieve 1 without prior.

E. Case Study

In Fig. 11, we give an illustrative example of our framework on political books network in which nodes represent books about U.S. politics sold by Amazon, and edges represent frequent co-purchasing of books by the same buyers. According to their political viewpoints, these books are divided into three categories: 1) "liberal"; 2) "neutral"; or 3) "conservative." In Fig. 11, we use the shape to represent the ground-truth types of the books, color to represent the results from our framework with different prior percentages. We use the percentage of misclassified nodes as an intuitive visual metric to judge the results. Assuming the co-purchasing books are more likely to have similar politics viewpoints, one aim to divide the books into three categories by using only topology information. However, the network structure is not very clear and the methods only based on network topology have the following drawbacks.

1) We can find only two densely connected communities using the link information, but the real number of categories is three. The reason may be that the third community, neutral books, is the overlapping part between the conservative and liberal books communities. Since the neutral books community do not have clear structure, i.e., these nodes do not densely connect with each other, it is insufficient to accurately determine communities by only using topology information, and the help of prior information is necessary.
2) From the result of NMF_LSE without prior as shown in Fig. 11(a), we find out that some nodes cannot be correctly classified only with the topology information. For example, the book *Power Plays* (node 47) connects



(a) (b) (c)

Fig. 11. Illustrative example of our framework on U.S. political books network. The three plots are the results from NMF_LSE with 0%, 10%, and 20% priors, respectively. In each plot, the shapes "circle ○," "square □," and "triangle △" represent the ground-truth communities are conservative, liberal, and neutral books, respectively. The colors "brown," "blue," and "green" represent the estimated communities are conservative, liberal, and neutral books using our framework, respectively. If the color of one node does not match its ground-truth shape, the node is not classified correctly by the method. (a) Baseline, misclassification: 18/105. (b) 10% priors, misclassification: 7/105. (c) 20% priors, misclassification: 0/105.

with three conservative books *Arrogance* (node 48), *A National Party No More* (node 9), and *Off With Their Heads* (node 13); the book *Buck Up Suck Up* (node 103) connects with three liberal books *We're Right They're Wrong* (node 96), *Had Enough?* (node 94), and *It's Still the Economy, Stupid!* (node 95); and also they connect with each other. But as we know in the ground-truth, the book *Buck Up Suck Up* (node 103) is a liberal book while the book *Power Plays* (node 47) is a neutral one. But the original NMF_LSE treats the *Buck Up Suck Up* as a liberal book and *Power Plays* as a conservative one, which is obviously not correct. This phenomenon also implies that topology information is not adequate to correctly classify in network with vague community structure and prior information is helpful.

3) It is hard for original methods to determine the boundaries of communities. For example, the nodes of the lowermost portion in Fig. 11(a) are misclassified by all original methods. Prior information is needed for solving the boundary problem.

In Fig. 11, we plot the results from our framework based on NMF_LSE. And the three figures are the results with 0%, 10%, and 20% of priors, respectively. As we can see, with the increase of the prior, the percentage of mismatching nodes decreases significantly in terms of accuracy. Besides, as we add 10% priors the boundary between liberal books community and neutral books community becomes clear as shown in Fig. 11(b). Furthermore, as the used priors reach 20%, the boundaries between conservative books community and neutral books community become clear and the neutral books community can be accurately detected as shown in Fig. 11(c). In conclusion, with the increase of used priors, the three problems mentioned above can be gradually solved.

VI. Conclusion

In this paper, we have provided a unified interpretation to a group of existing community detection algorithms, i.e., clustering in the latent space of nodes. And then we propose a unified semi-supervised framework based on latent space similarity, which combines the network topology with prior information using graph regularization. The proposed semi-supervised framework is applicable to any matrix based community detection algorithms as far as they can be interpreted using our unified interpretation, such as NMF, SC, and their variants. Different from previous works which transfer the semi-supervised community detection problem into the traditional community detection ones by directly modifying the adjacency matrix, we formulate it as a unified problem and balance the contributions of topology information and prior information in a seamless way. Extensive experiments on artificial and real networks illustrate the robustness and effectiveness of our framework on encoding prior information.

In the future, we may conduct research in the following two directions. Firstly, it would be interesting to investigate the structure of the prior information and design corresponding algorithms, e.g., approach to combine various types of priors from different sources, and online semi-supervised framework for sequently arriving prior information. Secondly, we will investigate how to design parallel algorithms to make our framework more efficient on large-scale networks and how to directly apply our semi-supervised framework to some existing efficient community detection algorithms for large networks.

References

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

[2] S. Wasserman, *Social Network Analysis: Methods and Applications*, vol. 8. Cambridge, U.K.: Cambridge Univ. Press, 1994.

[3] H. Lu *et al.*, "The interactome as a tree—An attempt to visualize the protein–protein interaction network in yeast," *Nucleic Acids Res.*, vol. 32, no. 16, pp. 4804–4811, 2004.

[4] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Math.*, vol. 6, no. 1, pp. 29–123, 2009.

[5] M. E. Newman, "Detecting community structure in networks," *Eur. Phys. J. B Condens. Matt. Complex Syst.*, vol. 38, no. 2, pp. 321–330, 2004.

[6] M. E. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, 2006.

[7] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, no. 3, pp. 75–174, 2010.

[8] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Phys. Rep.*, vol. 533, no. 4, pp. 95–142, 2013.

[9] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using Bayesian non-negative matrix factorization," *Phys. Rev. E*, vol. 83, no. 6, 2011, Art. ID 066114.

[10] R.-S. Wang, S. Zhang, Y. Wang, X.-S. Zhang, and L. Chen, "Clustering complex networks and biological networks by nonnegative matrix factorization with various similarity measures," *Neurocomputing*, vol. 72, no. 1, pp. 134–141, 2008.

[11] S. Garruzzo and D. Rosaci, "Agent clustering based on semantic negotiation," *ACM Trans. Auton. Adapt. Syst.*, vol. 3, no. 2, pp. 7:1–7:40, May 2008.

[12] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *Proc. 3rd ACM Int. Workshop Link Disc.*, Chicago, IL, USA, 2005, pp. 36–43.

[13] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, "Inference and phase transitions in the detection of modules in sparse networks," *Phys. Rev. Lett.*, vol. 107, Aug. 2011, Art. ID 065701.

[14] R. R. Nadakuditi and M. E. J. Newman, "Graph spectra and the detectability of community structure in networks," *Phys. Rev. Lett.*, vol. 108, May 2012, Art. ID 188701.

[15] F. Radicchi, "Driving interconnected networks to supercriticality," *Phys. Rev. X*, vol. 4, Apr. 2014, Art. ID 021014.

[16] F. Radicchi, "Detectability of communities in heterogeneous networks," *Phys. Rev. E*, vol. 88, Jul. 2013, Art. ID 010801.

[17] F. Radicchi, "A paradox in community detection," *Europhys. Lett.*, vol. 106, no. 3, 2014, Art. ID 38001.

[18] E. Eaton and R. Mansbach, "A spin-glass model for semi-supervised community detection," in *Proc. AAAI Conf. Artif. Intell.*, Toronto, ON, Canada, 2012, pp. 900–906.

[19] G. Ver Steeg, A. Galstyan, and A. E. Allahverdyan, "Statistical mechanics of semi-supervised clustering in sparse graphs," *J. Stat. Mech. Theory Exp.*, vol. 2011, no. 8, 2011, Art. ID P08009.

[20] A. E. Allahverdyan, G. Ver Steeg, and A. Galstyan, "Community detection with and without prior information," *Europhys. Lett.*, vol. 90, no. 1, 2010, Art. ID 18002.

[21] X. Ma, L. Gao, X. Yong, and L. Fu, "Semi-supervised clustering algorithm for community structure detection in complex networks," *Phys. A Stat. Mech. Appl.*, vol. 389, no. 1, pp. 187–197, 2010.

[22] Z.-Y. Zhang, "Community structure detection in complex networks with partial background information," *Europhys. Lett.*, vol. 101, no. 4, 2013, Art. ID 48005.

[23] Z.-Y. Zhang, K.-D. Sun, and S.-Q. Wang, "Enhanced community structure detection in complex networks with partial background information," *Sci. Rep.*, vol. 3, Nov. 2013, Art. ID 3241.

[24] 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.

[25] B. Yang, J. Liu, and D. Liu, "Characterizing and extracting multiplex patterns in complex networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 469–481, Apr. 2012.

[26] C. Liu, J. Liu, and Z. Jiang, "A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2274–2287, Dec. 2014.

[27] T. Leal, A. Goncalves, V. Da F Vieira, and C. Xavier, "DECoDe—Differential evolution algorithm for community detection," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Manchester, U.K., Oct. 2013, pp. 4635–4640.

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

[29] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, 2006, Art. ID 036104.

[30] C. Shi, Z. Yan, Y. Cai, and B. Wu, "Multi-objective community detection in complex networks," *Appl. Soft Comput.*, vol. 12, no. 2, pp. 850–859, 2012.

[31] P. De Meo, E. Ferrara, D. Rosaci, and G. Sarne, "Trust and compactness in social network groups," *IEEE Trans. Cybern.*, to be published.

[32] D. S. Bassett *et al.*, "Robust detection of dynamic community structure in networks," *Chaos Interdiscipl. J. Nonlin. Sci.*, vol. 23, Mar. 2013, Art. ID 013142.

[33] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen, "Quantitative function for community detection," *Phys. Rev. E*, vol. 77, no. 3, 2008, Art. ID 036109.

[34] S. Zhang, R.-S. Wang, and X.-S. Zhang, "Uncovering fuzzy community structure in complex networks," *Phys. Rev. E*, vol. 76, no. 4, 2007, Art. ID 046103.

[35] D. Wang, T. Li, S. Zhu, and C. Ding, "Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization," in *Proc. ACM 31st Annu. Int. ACM SIGIR Conf.*, Singapore, 2008, pp. 307–314.

[36] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 2000, pp. 556–562.

[37] T. N. Bui and C. Jones, "Finding good approximate vertex and edge partitions is NP-hard," *Inf. Process. Lett.*, vol. 42, no. 3, pp. 153–159, 1992.

[38] H.-W. Shen and X.-Q. Cheng, "Spectral methods for the detection of network community structure: A comparative analysis," *J. Stat. Mech. Theory Exp.*, vol. 2010, no. 10, 2010, Art. ID P10020.

[39] S. M. van Dongen, "Graph clustering by flow simulation," Ph.D. dissertation, Dutch Nat. Res. Inst. Math. Comput. Sci., University of Utrecht, Utrecht, The Netherlands, 2000.

[40] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[41] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *J. Symb. Comput.*, vol. 9, no. 3, pp. 251–280, 1990.

[42] M. R. Hestenes and E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems*, vol. 49. Washington, DC, USA: NBS, 1952.

[43] J. W. Demmel, M. T. Heath, and H. A. Van Der Vorst, "Parallel numerical linear algebra," *Acta Numer.*, vol. 2, pp. 111–197, 1993.

[44] C. Liu, H.-C. Yang, J. Fan, L.-W. He, and Y.-M. Wang, "Distributed nonnegative matrix factorization for web-scale dyadic data analysis on MapReduce," in *Proc. 19th Int. Conf. World Wide Web*, New York, NY, USA, 2010, pp. 681–690.

[45] M. Newman, "Spectral methods for network community detection and graph partitioning," *Phys. Rev. E*, vol. 88, Jul. 2013, Art. ID 042822.

[46] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, Jun. 2008.

[47] W. Zachary, "An information flow model for conflict and fission in small groups1," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, 1977.

[48] D. Lusseau and M. E. Newman, "Identifying the role that animals play in their social networks," *Proc. Roy. Soc. London B Biol. Sci.*, vol. 271, no. 6, pp. S477–S481, 2004.

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

[50] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.

[51] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E*, vol. 78, no. 4, 2008, Art. ID 046110.

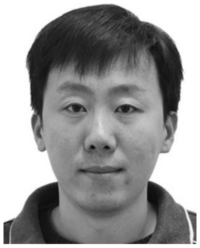
Liang Yang received the B.E. and M.E. degrees in computational mathematics from Nankai University, Tianjin, China, in 2004 and 2007, respectively. He is currently pursuing the Ph.D. degree from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

He is an Assistant Professor with the School of Information Engineering, Tianjin University of Commerce, Tianjin. His current research interests include community detection, semi-supervised learning, low-rank modeling, and deep learning.

Xiaochun Cao (SM'14) received the B.E. and M.E. degrees from Beihang University, Beijing, China, and the Ph.D. degree from the University of Central Florida, Orlando, FL, USA, all in computer science.

He was a Research Scientist at ObjectVideo, Inc., Reston, VA, USA, for three years. From 2008 to 2012, he was a Professor at Tianjin University, Tianjin, China. He is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing. He has authored and co-authored over 80 journal and conference papers.

Mr. Cao was the recipient of the Piero Zamperoni Best Student Paper Award at the International Conference on Pattern Recognition in 2004 and 2010 and nominated for the university level Outstanding Dissertation Award for his dissertation.



Di Jin received the B.S., M.S., and Ph.D. degrees in computer science from Jilin University, Changchun, China, in 2005, 2008, and 2012, respectively.

He was a Post-Doctoral Research Fellow at the School of Design, Engineering, and Computing, Bournemouth University, Poole, U.K., from 2013 to 2014. He is an Assistant Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. He has published over 30 international journal articles and conference papers. His current research interests include data mining, complex network analysis, and machine learning.

Xiao Wang was born in China, in 1988. He received the B.E. degree from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2009, and the M.E. degree from Henan University, Kaifeng, China, in 2012. He is currently pursuing the Ph.D. degree from the School of Computer Science and Technology, Tianjin University, Tianjin, China.

His current research interests include complex network, data mining, and machine learning.

Dan Meng (M'02) received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1995.

He is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His current research interests include high performance computing and computer architecture.

Prof. Meng is a Senior Member of china computer federation.