Adaptive Affinity Learning for Accurate Community Detection

Conference Paper · November 2018

DOI: 10.1109/ICDM.2018.00188

CITATIONS
8

READS
172

5 authors, including:

Chuan Chen
Sun Yat-Sen University
73 PUBLICATIONS **714** CITATIONS
SEE PROFILE

Zibin Zheng
Sun Yat-Sen University
372 PUBLICATIONS **15,797** CITATIONS
SEE PROFILE

Adaptive Affinity Learning for Accurate Community Detection

Fanghua Ye*†, Shenghui Li*†, Zhiwei Lin*†, Chuan Chen*†, Zibin Zheng*†
*School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
†National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China
{yefh5, lishh53, linzhw25}@mail2.sysu.edu.cn; {chenchuan, zhzibin}@mail.sysu.edu.cn

***Abstract*—The task of community detection has become a fundamental research problem in complex network analysis. Intuitively, similar nodes are more likely to be contained in the same community. However, most existing community detection methods cannot extract the intrinsic similarity between nodes. Thus, they may fail to identify the real community structures. In this paper, we propose to learn an affinity matrix adaptively, which can capture the intrinsic similarity between nodes accurately, and therefore benefit the community detection results. Specifically, the proposed model first embeds each node into a low-dimensional space through a transformation matrix with the community structures being preserved. Then, our model learns the affinity matrix in this low-dimensional space. The affinity matrix is further utilized to guide the learning of the community membership matrix via manifold regularization. The above three matrices are learned simultaneously and updated iteratively under the framework of Alternating Direction Method of Multipliers (ADMM). Extensive experiments show that our model can outperform the state-of-the-art approaches.**

***Keywords*-community detection, affinity learning, nonnegative matrix factorization, complex networks, ADMM**

I. Introduction

Community detection targets at assigning community labels to nodes in a given network such that the nodes in the same community share higher similarity than the nodes in different communities [1]. Over the last two decades, lots of community detection methods have been proposed [2]. These methods can be roughly categorized into two classes: 1) The methods [3], [4] that can automatically identify the optimal community structures based on a certain criterion, e.g., modularity [5] and permanence [6]; 2) The methods [7], [8], [9] that require a parameter to specify the number of communities to detect. The first class of methods usually seek to identify communities by just leveraging the explicit link information. Typical methods of this class include Girvan-Newman's modularity maximization algorithm [1] and its variants like Louvain [4]. However, the link topology is a microscopic description of the whole network, it cannot reflect the intrinsic community-level similarity between nodes accurately due to its lower granularity. As a result, the detected communities may be quite different from the ground-truth. Recently, the second class of methods are gaining increasing popularity, among which, the nonnegative matrix factorization (NMF) based model has been widely adopted [7], [8], [9]. This is because NMF has high interpretability derived from the nonnegative constraints and more information can be incorporated into the framework of NMF flexibly via manifold regularization [10]. Obviously, the additional information can be used to better describe the similarity between nodes. However, the additional information is not always available and usually application-oriented and one-sided, thus it is still quite difficult to reflect the intrinsic similarity between nodes.

In this paper, we propose a novel framework to adaptively learn an affinity matrix with the ability to capture the intrinsic similarity between nodes, and therefore lead to high-quality community detection results. The proposed community detection model is built upon the NMF framework, and is named as Adaptive Affinity NMF (A^{2}NMF). Instead of taking more additional information into consideration, we aim to make the best of the network topology. Thus, our model is more general and more applicable. It is noted that networks gathered from real world are inevitably incomplete, i.e., there may be missing or noisy edges/links. Therefore, in order to make the learned affinity matrix more robust, we first embed each node into a low-dimensional space via a transformation matrix, which can preserve the community structures well. We then learn the affinity matrix in this low-dimensional space. And the learned affinity matrix is further employed to guide the learning of the community membership matrix. These three matrices are learned simultaneously rather than separately, thus they can guide each other during the learning process. This mutual guidance makes our model capable of finding the intrinsic similarity between nodes and the accurate community memberships. Extensive experiments further demonstrate the efficiency and effectiveness of the proposed model.

II. Revisiting NMF for Community Detection

Notations: Throughout this paper, we denote matrices by bold uppercase letters and vectors by bold lowercase letters. For a given matrix $\mathbf{X}$, its i-th row and j-th column are denoted by $\mathbf{x}^i$ and $\mathbf{x}_j$, and its (i,j)-th element is denoted by x_{ij}. The trace and Frobenius norm of $\mathbf{X}$ are denoted by $\mathrm{Tr}(\mathbf{X})$ and $\|\mathbf{X}\|_F$ respectively. Then, we have $\|\mathbf{X}\|_F = \sqrt{\mathrm{Tr}(\mathbf{X}^T\mathbf{X})}$. Besides, we use $< \mathbf{X}, \mathbf{Y} >$ to denote the Euclidean inner product between matrices $\mathbf{X}$ and $\mathbf{Y}$, i.e., $< \mathbf{X}, \mathbf{Y} >= \mathrm{Tr}(\mathbf{X}^T\mathbf{Y})$. In particular, we use $\mathbf{I}_c$ to denote the identity matrix of size $c \times c$. The ℓ_2-norm of a vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\|_2$.

Let $G = (V, E)$ be an undirected and unweighted simple network with $n = |V|$ nodes and $m = |E|$ edges, where V and E denote the node set and edge set respectively. Network

978-1-5386-9159-5/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDM.2018.00188

G is typically represented by a binary-valued adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$ whose (i,j)-th element a_{ij} indicates whether there is an edge between nodes i and j or not. The problem of community detection is to partition network G into k disjoint subnetworks such that nodes in each subnetwork are densely connected (highly similar) and nodes across different subnetworks are sparsely connected (highly dissimilar). Each subnetwork corresponds to a community.

In general, the NMF-based community detection methods seek to factorize the adjacency matrix $\mathbf{A}$ into the product of two identical low-rank matrices with the nonnegative constraints, i.e., $\mathbf{A} \approx \mathbf{U}\mathbf{U}^T (\mathbf{U} \in \mathbb{R}_+^{n\times k})$. Each row of $\mathbf{U}$ denotes a node, and each column of $\mathbf{U}$ represents a community. Therefore, u_{ij} can be interpreted as the probability of node i belonging to community j. From a generative model perspective, u_{ij} can be further treated as the probability that node i generates an edge belonging to community j. Accordingly, a_{ir} can be viewed as the probability that there exists an edge between nodes i and r in network G. Consequently, the probability that nodes i and r are linked in community j can be calculated by $u_{ij}u_{rj}$. Summing over all communities, we obtain the probability that nodes i and r are linked in network G is $\sum_{j=1}^{k} u_{ij}u_{rj}$, which should be as consistent as possible with a_{ir}. Thus, we have the following objective function:

$$\min_{\mathbf{U}} \ \|\mathbf{A} - \mathbf{U}\mathbf{U}^T\|_F^2, \ \text{s.t.} \ \mathbf{U} \geq 0. \tag{1}$$

Based on the learned $\mathbf{U}$, the community membership of each node i is defined as the index of the largest element in $\mathbf{u}^i$.

The model in (1) aims to minimize the reconstruction error of $\mathbf{A}$ from $\mathbf{U}$. Clearly, its performance depends heavily on the completeness of $\mathbf{A}$. When there are missing or noisy edges, the performance may degrade severely. To make the model in (1) more robust, extra constraints should be imposed on $\mathbf{U}$. One of the most widely adopted methods is to add a manifold regularization term. For community detection, the most basic assumption is that two linked nodes are more similar than two non-linked nodes [9]. Thus, we can derive:

$$\min_{\mathbf{U}} \ \|\mathbf{A} - \mathbf{U}\mathbf{U}^T\|_F^2 + \gamma \mathrm{Tr}(\mathbf{U}^T\mathbf{L}_\mathbf{A}\mathbf{U}), \ \text{s.t.} \ \mathbf{U} \geq 0, \tag{2}$$

where γ is the regularization parameter and $\mathbf{L}_\mathbf{A}$ is the Laplacian matrix defined by $\mathbf{D}_\mathbf{A} - \mathbf{A}$. Here, $\mathbf{D}_\mathbf{A}$ is a diagonal matrix whose elements are row sums of $\mathbf{A}$.

However, in (2), the Laplacian matrix $\mathbf{L}_\mathbf{A}$ is calculated based on the adjacency matrix $\mathbf{A}$ directly. There are two main drawbacks: 1) The quality of $\mathbf{A}$ still affects the performance significantly; 2) Since $\mathbf{A}$ is binary-valued, it is unable to distinguish the similarity of different linked node pairs. Therefore, it is a necessity to extract the intrinsic similarity between nodes to better fit the community detection task.

III. A^{2}NMF: The Proposed Model

In this paper, we aim to learn adaptively an affinity matrix $\mathbf{S} \in \mathbb{R}_+^{n\times n}$ with each element s_{ij} denoting the learned similarity between nodes i and j. $\mathbf{S}$ is expected to better describe the intrinsic similarity between nodes, and consequently more accurate community detection results can be obtained.

A. Adaptive Affinity Learning

It is reasonable to characterize a node by its linked nodes and its non-linked nodes. Therefore, we denote the adjacency matrix $\mathbf{A}$ as $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n]$. In this way, each column of $\mathbf{A}$ is regarded as a feature vector of the corresponding node. To learn the affinity matrix $\mathbf{S}$, a basic assumption is that nodes with smaller distance should have larger similarity value. It is straightforward to fulfill the basic assumption by calculating the distance between nodes i and j as $\|\mathbf{a}_i - \mathbf{a}_j\|_2$. Unfortunately, this simple measurement will result in unsatisfactory performance. Due to the sparsity of real-world networks (i.e., most elements of $\mathbf{a}_i$ or $\mathbf{a}_j$ are zeros), it is inappropriate to calculate the distance between nodes i and j as $\|\mathbf{a}_i - \mathbf{a}_j\|_2$ directly. Besides, this simple measurement still suffers from the effects of missing or noisy edges. To address these issues, we introduce a transformation matrix $\mathbf{Q} \in \mathbb{R}^{n\times k'} (k' \ll n)$ to embed each node into the low-dimensional space $\mathbb{R}^{k'}$. In this low-dimensional space $\mathbb{R}^{k'}$, the distance between nodes i and j is then calculated as $\|\mathbf{Q}^T\mathbf{a}_i - \mathbf{Q}^T\mathbf{a}_j\|_2$, which can reduce the effects of both network sparsity and network noises. The introduction of $\mathbf{Q}$ is inspired by Locality Preserving Projection (LPP) [11], which requires the local similarity among different nodes should be preserved. In this regard, $\mathbf{Q}$ plays the role of dimensionality reduction. With the aid of $\mathbf{Q}$, it is clear that the sparsity issue and the effects of network noises can be reduced. Then, we learn the affinity matrix $\mathbf{S}$ adaptively by:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{Q}} \ \sum_{i,j=1}^{n} \|\mathbf{Q}^T\mathbf{a}_i - \mathbf{Q}^T\mathbf{a}_j\|_2^2 s_{ij} + \alpha\|\mathbf{S}\|_F^2, \\ &\text{s.t.} \ \ \mathbf{Q}^T\mathbf{A}\mathbf{A}^T\mathbf{Q} = \mathbf{I}_{k'}, \forall i, \sum_{j=1}^{n} s_{ij} = 1, s_{ij} \geq 0, s_{ii} = 0, \end{aligned} \tag{3}$$

where α is a positive regularization parameter. Different from LPP that imposes the orthogonality constraint on $\mathbf{Q}$ directly [11], we choose to enforce the orthogonality constraint on $\mathbf{Q}^T\mathbf{A}$, which can help strengthen the independence of different feature dimensions in $\mathbb{R}^{k'}$. The summation constraint on each row of $\mathbf{S}$ enables the learned similarity to be shift invariant. The constraint $s_{ii} = 0$ is employed to avoid self-loop.

In (3), $\mathbf{S}$ is not restricted to be necessarily symmetric, this is because similarity is usually non-isotropic across the networks [12]. This relaxation also makes our model more general and much easier to optimize. Based on the learned $\mathbf{S}$ in (3), we can construct the Laplacian matrix $\mathbf{L}_\mathbf{S}$ as $\mathbf{L}_\mathbf{S} = \mathbf{D}_\mathbf{S} - (\mathbf{S}^T + \mathbf{S})/2$, here, $\mathbf{D}_\mathbf{S}$ is a diagonal matrix whose elements are row sums of $(\mathbf{S}^T + \mathbf{S})/2$. Clearly, $\mathbf{L}_\mathbf{S}$ is guaranteed to be positive semidefinite. Then $\mathbf{L}_\mathbf{S}$ can be used to guide the learning of $\mathbf{U}$. However, this two-stage strategy may result in unsatisfactory performance. This is because in the optimization of (3), there is no guidance information with respect to the community structures. To solve this problem, we choose to learn $\mathbf{U}$ and $\mathbf{S}$ simultaneously. It is straightforward to achieve this goal by replacing $\mathbf{L}_\mathbf{A}$ with $\mathbf{L}_\mathbf{S}$ in (2) and combining (2) and (3) into a unified objective function. In this way, $\mathbf{U}$ can guide the learning of $\mathbf{S}$ via the regularization term $\mathrm{Tr}(\mathbf{U}^T\mathbf{L}_\mathbf{S}\mathbf{U})$.

B. Community Preserving Node Embedding

Denote the network associated with $\mathbf{S}$ as $G' = (V, E')$, whose node set is the same as that of G. Since $\mathbf{S}$ is expected to capture the intrinsic similarity between nodes more accurately, G' should have explicit community structures. Ideally, G' should have exactly k connected components. In graph theory, we have the following theorem [13], [14]:

Theorem 1. *The multiplicity k of the eigenvalue 0 of the Laplacian matrix $\mathbf{L_S}$ equals the number of connected components in the network G'.*

Let δ_i denote the i-th smallest eigenvalue of $\mathbf{L_S}$. Since $\mathbf{L_S}$ is positive semidefinite, we have $\delta_i \geq 0$ for each i. According to Theorem 1, G' having k connected components indicates that $\sum_{i=1}^{k} \delta_i = 0$ [14]. Recall that $\mathbf{S}$ is learned in the low-dimensional space $\mathbb{R}^{k'}$, thus it is essential to make nodes contained in the same community group together in this low-dimensional space. To this end, we derive the following objective function according to the Ky Fan's Theorem [15]:

$$\sum_{i=1}^{k} \delta_i = \min_{\mathbf{Q}^T\mathbf{A}\mathbf{A}^T\mathbf{Q}=\mathbf{I}_k} \mathrm{Tr}(\mathbf{Q}^T\mathbf{A}\mathbf{L_S}\mathbf{A}^T\mathbf{Q}). \tag{4}$$

Note that (4) corresponds to the first term in (3). To make $\mathbf{S}$ better fit for community detection, (4) shows that we should set $k' = k$. That is, each node is embedded into the low-dimensional space $\mathbb{R}^k$. By optimizing (4), the nodes belonging to the same community will be grouped together in the space $\mathbb{R}^k$. Thus, when $k' = k$, the optimal transformation matrix $\mathbf{Q}$ is able to preserve the community structures well.

C. The Unified Model

Based on the above discussions, we can now derive the final objective function of our proposed A^{2}NMF model as follows:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{S},\mathbf{Q}} \ & \|\mathbf{A} - \mathbf{U}\mathbf{U}^T\|_F^2 + \gamma \mathrm{Tr}(\mathbf{U}^T\mathbf{L_S}\mathbf{U}) \\ & + \beta \sum_{i,j=1}^{n} \|\mathbf{Q}^T\mathbf{a}_i - \mathbf{Q}^T\mathbf{a}_j\|_2^2 s_{ij} + \alpha\|\mathbf{S}\|_F^2, \\ \text{s.t.} \ & \mathbf{Q}^T\mathbf{A}\mathbf{A}^T\mathbf{Q} = \mathbf{I}_k, \mathbf{U}^T\mathbf{U} = \mathbf{I}_k, \mathbf{U} \geq 0, \\ & \forall i, \sum_{j=1}^{n} s_{ij} = 1, 0 \leq s_{ij} \leq 1, s_{ii} = 0, \end{aligned} \tag{5}$$

where β is a tuning parameter. Note that we have imposed the orthogonality constraint on $\mathbf{U}$ as well, which has two benefits: 1) With this constraint, some trivial solutions can be avoided; 2) The obtained $\mathbf{U}$ will be sparse, thus can better reflect the community memberships.

IV. Optimization

The objective function in (5) is not convex over the three variables, i.e., $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{Q}$. To solve (5) efficiently, we consider to employ the framework of Alternating Direction Method of Multipliers (ADMM) [16] to optimize the three variables iteratively. By further introducing two auxiliary variables $\mathbf{V} = \mathbf{U}$ and $\mathbf{Z} = \mathbf{Q}^T\mathbf{A}$, then, we can rewrite the objective function in (5) into the following equivalent objective function:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{S},\mathbf{Q},\mathbf{V},\mathbf{Z}} & \|\mathbf{A} - \mathbf{V}\mathbf{U}^T\|_F^2 + \gamma \mathrm{Tr}(\mathbf{U}^T\mathbf{L_S}\mathbf{U}) \\ & + \beta \sum_{i,j=1}^{n} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 s_{ij} + \alpha\|\mathbf{S}\|_F^2, \\ \text{s.t.} \ & \mathbf{V} = \mathbf{U}, \mathbf{Z} = \mathbf{Q}^T\mathbf{A}, \mathbf{Z}\mathbf{Z}^T = \mathbf{I}_k, \mathbf{V}^T\mathbf{V} = \mathbf{I}_k, \\ & \mathbf{U} \geq 0, \forall i, \sum_{j=1}^{n} s_{ij} = 1, 0 \leq s_{ij} \leq 1, s_{ii} = 0. \end{aligned} \tag{6}$$

By introducing the augmented Lagrangian multipliers $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, and μ, (6) becomes:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{S},\mathbf{Q},\mathbf{V},\mathbf{Z}} & \|\mathbf{A} - \mathbf{V}\mathbf{U}^T\|_F^2 + \gamma \mathrm{Tr}(\mathbf{U}^T\mathbf{L_S}\mathbf{U}) \\ & + \beta \sum_{i,j=1}^{n} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 s_{ij} + \alpha\|\mathbf{S}\|_F^2 \\ & + < \boldsymbol{\lambda}_1, \mathbf{V} - \mathbf{U} > + < \boldsymbol{\lambda}_2, \mathbf{Z} - \mathbf{Q}^T\mathbf{A} > \\ & + \frac{\mu}{2}(\|\mathbf{V} - \mathbf{U}\|_F^2 + \|\mathbf{Z} - \mathbf{Q}^T\mathbf{A}\|_F^2), \\ \text{s.t.} \ & \mathbf{Z}\mathbf{Z}^T = \mathbf{I}_k, \mathbf{V}^T\mathbf{V} = \mathbf{I}_k, \mathbf{U} \geq 0, \\ & \forall i, \sum_{j=1}^{n} s_{ij} = 1, 0 \leq s_{ij} \leq 1, s_{ii} = 0. \end{aligned} \tag{7}$$

In the ADMM framework, (7) is divided into the following five sub-problems.

Update U: With $\mathbf{S}$, $\mathbf{Q}$, $\mathbf{V}$, $\mathbf{Z}$ fixed, the objective function in (7) is reduced to:

$$\begin{aligned} \min_{\mathbf{U} \geq 0} \ & \|\mathbf{A} - \mathbf{V}\mathbf{U}^T\|_F^2 + \gamma \mathrm{Tr}(\mathbf{U}^T\mathbf{L_S}\mathbf{U}) \\ & + < \boldsymbol{\lambda}_1, \mathbf{V} - \mathbf{U} > + \frac{\mu}{2}\|\mathbf{V} - \mathbf{U}\|_F^2. \end{aligned} \tag{8}$$

Given the fact that $\mathbf{V}^T\mathbf{V} = \mathbf{I}_k$, (8) can be further reduced as:

$$\min_{\mathbf{U} \geq 0} \ \frac{1}{2}\mathrm{Tr}(\mathbf{U}^T\mathbf{R}\mathbf{R}^T\mathbf{U}) - < \mathbf{P}, \mathbf{U} >, \tag{9}$$

where

$$\begin{cases} \mathbf{R}\mathbf{R}^T = 2\gamma\mathbf{L_S} + (\mu + 2)\mathbf{I}_n, \\ \mathbf{P} = \boldsymbol{\lambda}_1 + \mu\mathbf{V} + 2\mathbf{A}\mathbf{V}. \end{cases} \tag{10}$$

It is easy to see that (9) is equivalent to the following Non-Negative Least Squares (NNLS) problem:

$$\min_{\mathbf{U} \geq 0} \ \|\mathbf{R}^T\mathbf{U} - \mathbf{R}^{-1}\mathbf{P}\|_F^2, \tag{11}$$

which can be solved by the algorithm proposed in [17].

Update S: When $\mathbf{U}$, $\mathbf{Q}$, $\mathbf{V}$, $\mathbf{Z}$ are fixed, we have:

$$\begin{aligned} \min_{\mathbf{S}} \ & \gamma \mathrm{Tr}(\mathbf{U}^T\mathbf{L_S}\mathbf{U}) + \beta \sum_{i,j=1}^{n} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 s_{ij} + \alpha\|\mathbf{S}\|_F^2, \\ \text{s.t.} \ & \forall i, \sum_{j=1}^{n} s_{ij} = 1, 0 \leq s_{ij} \leq 1, s_{ii} = 0. \end{aligned} \tag{12}$$

Note that $\mathrm{Tr}(\mathbf{U}^T\mathbf{L_S}\mathbf{U}) = \frac{1}{4}\sum_{i,j=1}^{n} \|\mathbf{u}^i - \mathbf{u}^j\|_2^2 (s_{ij} + s_{ji}) = \frac{1}{2}\sum_{i,j=1}^{n} \|\mathbf{u}^i - \mathbf{u}^j\|_2^2 s_{ij}$. Let $d_{ij}^u = \|\mathbf{u}^i - \mathbf{u}^j\|_2^2$ and $d_{ij}^z = \|\mathbf{z}_i - \mathbf{z}_j\|_2^2$. Therefore, (12) can be rewritten as follows:

$$\min_{\mathbf{S}} \sum_{i,j=1}^{n} \big(\frac{\gamma}{2} d_{ij}^{u} s_{ij} + \beta d_{ij}^{z} s_{ij} + \alpha s_{ij}^{2}\big), \quad \text{s.t. } \forall i, \sum_{j=1}^{n} s_{ij} = 1, 0 \le s_{ij} \le 1, s_{ii} = 0. \tag{13}$$

Note that (13) is independent of different is, thus we can optimize $\mathbf{S}$ row-by-row. Denote $\mathbf{d}^i \in \mathbb{R}^n$ as a row vector with the j-th element being $d_{ij} = -(\frac{\gamma}{2} d_{ij}^u + \beta d_{ij}^z)$. Then, for each i, (13) can be decomposed into the following sub-problem:

$$\min_{\mathbf{s}^i} \|\mathbf{s}^i - \frac{1}{2\alpha}\mathbf{d}^i\|_2^2, \text{ s.t. } \sum_{j=1}^{n} s_{ij} = 1, s_{ij} \ge 0, s_{ii} = 0. \tag{14}$$

Without loss of generality, we assume that all the elements in $\mathbf{d}^i$ are sorted in descending order, i.e., $d_{i1} \ge d_{i2} \ge \cdots \ge d_{in}$. Since $s_{ii} = 0$, we denote $\hat{\mathbf{s}}^i$ as $\hat{\mathbf{s}}^i = [s_{i1}, s_{i2}, \cdots, s_{i(i-1)}, s_{i(i+1)}, \cdots, s_{in}]$ (without s_{ii} in $\hat{\mathbf{s}}^i$) and $\hat{\mathbf{d}}^i$ as $\hat{\mathbf{d}}^i = [d_{i1}, d_{i2}, \cdots, d_{i(i-1)}, d_{i(i+1)}, \cdots, d_{in}]$ (without d_{ii} in $\hat{\mathbf{d}}^i$). Then, (14) is transformed into the following problem:

$$\min_{\hat{\mathbf{s}}^i} \|\hat{\mathbf{s}}^i - \frac{1}{2\alpha}\hat{\mathbf{d}}^i\|_2^2, \text{ s.t. } \sum_{j=1}^{n-1} \hat{s}_{ij} = 1, \hat{s}_{ij} \ge 0, \tag{15}$$

which can be efficiently solved by the method in [18]. Based on $\hat{\mathbf{s}}^i$, we can obtain the optimal solution of $\mathbf{s}^i$ immediately.

Update Q: To update $\mathbf{Q}$, we fix $\mathbf{U}$, $\mathbf{S}$, $\mathbf{V}$, $\mathbf{Z}$, and remove terms that are irrelevant to $\mathbf{Q}$. Then, we obtain:

$$\min_{\mathbf{Q}} \frac{\mu}{2}\|\mathbf{Z} - \mathbf{Q}^T\mathbf{A}\|_F^2 + < \boldsymbol{\lambda}_2, \mathbf{Z} - \mathbf{Q}^T\mathbf{A} > . \tag{16}$$

Obviously, (16) is quadratic and convex over $\mathbf{Q}$. Setting the derivative of (16) w.r.t. $\mathbf{Q}$ to zero, we have:

$$\mathbf{Q} = (\mathbf{A}\mathbf{A}^T)^{\dagger}\mathbf{A}\big(\mathbf{Z}^T + \frac{1}{\mu}\boldsymbol{\lambda}_2^T\big), \tag{17}$$

where $(\mathbf{A}\mathbf{A}^T)^{\dagger}$ denotes the Moore-Penrose inverse of $\mathbf{A}\mathbf{A}^T$.

Update V: By fixing $\mathbf{U}$, $\mathbf{S}$, $\mathbf{Q}$, $\mathbf{Z}$, the objective function in (7) is reduced to the sub-problem as below:

$$\min_{\mathbf{V}^T\mathbf{V}=\mathbf{I}_k} \|\mathbf{A} - \mathbf{V}\mathbf{U}^T\|_F^2 + < \boldsymbol{\lambda}_1, \mathbf{V} - \mathbf{U} > + \frac{\mu}{2}\|\mathbf{V} - \mathbf{U}\|_F^2. \tag{18}$$

The above sub-problem can be further rewritten as:

$$\min_{\mathbf{V}^T\mathbf{V}=\mathbf{I}_k} \frac{\mu}{2}\|\mathbf{V}\|_F^2 - \mu < \mathbf{H}, \mathbf{V} >, \tag{19}$$

where

$$\mathbf{H} = -\frac{1}{\mu}\boldsymbol{\lambda}_1 + \mathbf{U} + \frac{2}{\mu}\mathbf{A}\mathbf{U}. \tag{20}$$

Then, we arrive at:

$$\min_{\mathbf{V}^T\mathbf{V}=\mathbf{I}_k} \|\mathbf{V} - \mathbf{H}\|_F^2, \tag{21}$$

whose optimal solution can be easily generated as:

$$\mathbf{V} = \boldsymbol{\Omega}_1\boldsymbol{\Omega}_2^T, \tag{22}$$

TABLE I: Statistics of networks.

Networks	n	m	k
Polbooks	105	441	3
Football	115	613	12
PoliticsIE	348	12,567	7
PoliticsUK	419	19,950	5
Olympics	464	7,787	28
EmailEU	1,005	25,571	42
Polblogs	1,490	16,715	2

where $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$ are the left and right singular vectors of the economic Singular Value Decomposition (SVD) of $\mathbf{H}$.

Update Z: When updating $\mathbf{Z}$ with the other variables fixed, the objective function in (7) becomes:

$$\min_{\mathbf{Z}} 2\beta\mathrm{Tr}(\mathbf{Z}\mathbf{L}_{\mathbf{S}}\mathbf{Z}^T) + < \boldsymbol{\lambda}_2, \mathbf{Z} - \mathbf{Q}^T\mathbf{A} > + \frac{\mu}{2}\|\mathbf{Z} - \mathbf{Q}^T\mathbf{A}\|_F^2, \text{ s.t. } \mathbf{Z}\mathbf{Z}^T = \mathbf{I}_k, \tag{23}$$

which can be efficiently solved by the algorithm in [19].

Update Lagrangian Multipliers: In each iteration, we update the Lagrangian multipliers as follows:

$$\begin{aligned} \boldsymbol{\lambda}_1 &= \boldsymbol{\lambda}_1 + \mu(\mathbf{V} - \mathbf{U}), \\ \boldsymbol{\lambda}_2 &= \boldsymbol{\lambda}_2 + \mu(\mathbf{Z} - \mathbf{Q}^T\mathbf{A}), \\ \mu &= \rho\mu \ (\rho \ge 1). \end{aligned} \tag{24}$$

Until now, we have all the update rules done. By alternating the optimization of the five sub-problems above, the objective function in (7) is guaranteed to be nonincreasing, and finally converges to the optimal solution, which is benefited from the convergence guarantee of the ADMM framework.

V. Experimental Analysis

A. Dataset Description and Evaluation Metrics

Dataset Description: In the experiments, we adopt seven real-world networks as benchmark datasets. These networks are downloaded or extracted from three websites: Network Data[1] (Polbooks, Football, and Polblogs), Insight Project Resources[2] (PoliticsIE, PoliticsUK, and Olympics), and SNAP[3] (EmailEU). All the networks have ground-truth communities. Their basic information is listed in Tabel I.

Evaluation Metrics: Considering that all the networks have ground-truth communities, we employ Normalized Mutual Information (**NMI**) and Accuracy (**ACC**) as the evaluation metrics. Both NMI and ACC take value in the range of $[0, 1]$. And higher value indicates better community detection results. For their detailed descriptions, please refer to [6].

B. Comparison Methods

We choose eight representative methods as baseline methods, including two spectral clustering related methods, namely *Leading Eigenvector (LE)* [20] and *Constrained Laplacian Rank (CLR)* [14], and six NMF related methods, namely *Symmetric NMF (SNMF)* [7], *Graph regularized NMF (GNMF)*

[1] http://www-personal.umich.edu/~mejn/netdata/
[2] http://mlg.ucd.ie/index.html
[3] https://snap.stanford.edu/data/index.html

TABLE II: Performance comparison on NMI and ACC (bold numbers represent the best results).

	Networks	LE	CLR	SNMF	GNMF	BNMF	BigClam	HNMF	MNMF	A^{2}NMF
NMI	Polbooks	0.5281	0.5745	0.5253	0.5410	0.5195	0.4215	0.5253	0.0256	**0.6435**
	Football	0.6648	0.9242	0.9116	0.9127	0.9072	0.8273	0.8118	0.2518	**0.9385**
	PoliticsIE	0.7778	0.8395	0.7494	0.8259	0.7896	0.6580	0.5910	0.0345	**0.9058**
	PoliticsUK	0.8612	0.8809	0.7447	0.8752	0.7777	0.5411	0.6577	0.0165	**0.9720**
	Olympics	0.4924	0.8646	0.8505	0.8741	0.9077	0.8386	0.6226	0.2411	**0.9253**
	EmailEU	0.5196	0.5221	0.6839	**0.7025**	0.6763	0.5875	0.5083	0.1999	0.6977
	Polblogs	0.1238	0.0136	0.4493	0.4530	0.4955	0.1373	0.5012	0.0010	**0.5604**
ACC	Polbooks	0.7810	0.8381	0.7410	0.7671	0.7700	0.6890	0.7905	0.4052	**0.8667**
	Football	0.5478	0.9130	0.8917	0.9048	0.8696	0.7613	0.7913	0.2426	**0.9333**
	PoliticsIE	0.8592	0.8879	0.6865	0.8075	0.8546	0.7409	0.5948	0.2057	**0.9368**
	PoliticsUK	0.9498	0.9618	0.7288	0.9588	0.8982	0.7323	0.6993	0.2437	**0.9871**
	Olympics	0.3664	0.8254	0.7603	0.8093	0.8689	0.7161	0.5000	0.1422	**0.8980**
	EmailEU	0.3821	0.3950	0.5614	0.5974	0.5845	0.4839	0.3443	0.0978	**0.6359**
	Polblogs	0.4282	0.5067	0.8684	0.8698	0.8728	0.5939	0.8826	0.5156	**0.9007**

[21], *Bayesian NMF (BNMF)* [8], *BigClam* [22], *Homophily-based NMF (HNMF)* [9], and *Modularized NMF (MNMF)* [23]. The six NMF-based methods are highly comparable with our A^{2}NMF model.

C. Parameter Settings

Since all the eight baseline methods and our A^{2}NMF model require a parameter to specify the number of communities to detect, to test whether they can identify the ground-truth communities or not, the number of communities (i.e., k) to extract for each method is set to the number of ground-truth communities directly, as listed in Table I. For the methods LE, CLR, SNMF and BNMF, there are no more parameters to be set. For the methods BigClam and HNMF, we tune the parameters following the guidance of their authors. In order to identify the community structures of networks, MNMF needs to embed each node into a low-dimensional space. Here, we fix the dimensionality at 100 consistently. For GNMF and our A^{2}NMF model, we tune all the parameters in the range of $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$. Note that we tune the parameters in such a wide range, the purpose of which is to provide a comprehensive analysis of the performance of our model. However, as shown in our analyses of the parameters sensitivity (see details in Section V-E), the parameters can be tuned in a smaller range in practice. For a fair comparison, we run each method 20 times, and the average results are reported.

D. Community Detection Results

The results with respect to NMI and ACC are shown in Table II, where the bold numbers represent the best results. As can be seen, our A^{2}NMF model significantly outperforms the eight baseline methods on all the networks, except for performing the second best on EmailEU in term of NMI. Taking Polblogs as an example, we can see that this network is very sparse and it contains only two communities, which means that this network does not have significant community structures. Even so, our A^{2}NMF model achieves 90% accuracy in this network. For the evaluation metric NMI, our A^{2}NMF model also achieves over 5% performance promotion. These results verify that our A^{2}NMF model has strong ability to identify communities more accurately than the baseline methods. The success of our A^{2}NMF model lies in the fact that it can learn a robust affinity matrix which captures the intrinsic similarity between nodes. This affinity matrix then guides the learning of the community memberships more precisely.

E. Analysis

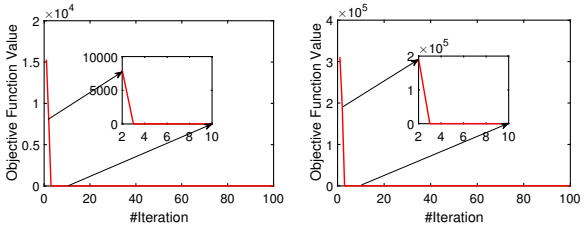
In this subsection, we further analyze the convergence speed and the parameters sensitivity of our A^{2}NMF model on the Polbooks and PoliticsUK networks. Similar results can be observed on the other five networks.

Convergence Analysis: To test the convergence speed, we fix all the parameters at 1. The results are shown in Fig. 1. As can be seen, our A^{2}NMF model converges very fast on both networks. The objective function value becomes stable within only a few iterations (usually less than 10). The results show that our optimization algorithm is very efficient, and it is able to converge to the optimal solution rapidly. Therefore, in practice, we can set the number of iterations to be 10.

Parameter Analysis: Recall that our A^{2}NMF model has three parameters to adjust the contribution of different components in (5). To analyze the sensitivity of the parameters α, β and γ, we tune each parameter at the range of $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$ and fix the other two parameters at 1. The results are shown in Fig. 2, Fig. 3 and Fig. 4 respectively. Note that we have adopted logarithmic scale for the x-axis in these three figures. It can be observed that our A^{2}NMF model is quite robust to parameters α and β, while it is sensitive to parameter γ. When γ becomes large, the reconstruction error of the adjacency matrix cannot be well controlled, which makes our model fail to identify the community structures.

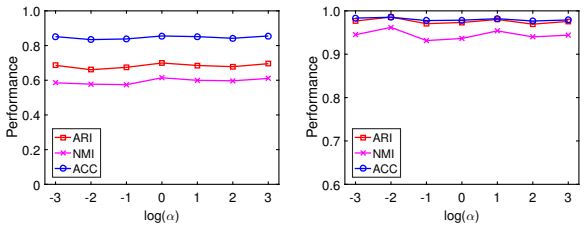
VI. Conclusion

In this paper, we propose A^{2}NMF to learn the community memberships and the affinity matrix that can capture the intrinsic similarity between nodes simultaneously. Due to the mutual guidance between the learning of the community membership matrix and the learning of the affinity matrix, we are able to find the intrinsic similarity between nodes and the accurate community memberships. To make our A^{2}NMF model more robust, we first embed each node into a low-dimensional space via a transformation matrix with community structures being



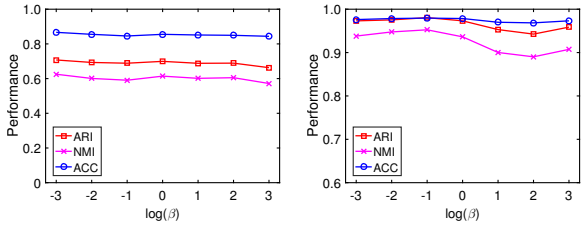
(a) Polbooks (b) PoliticsUK

Fig. 1: Convergence analysis of our A^{2}NMF model.



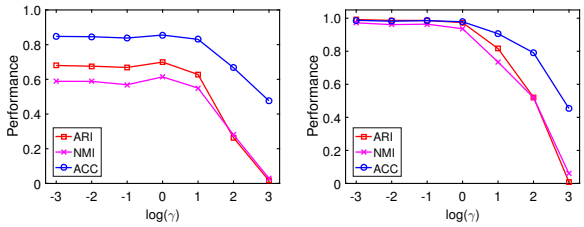
(a) Polbooks (b) PoliticsUK

Fig. 2: Effects of parameter α on the performance of A^{2}NMF.



(a) Polbooks (b) PoliticsUK

Fig. 3: Effects of parameter β on the performance of A^{2}NMF.



(a) Polbooks (b) PoliticsUK

Fig. 4: Effects of parameter γ on the performance of A^{2}NMF.

preserved. And then the affinity matrix is learned in this low-dimensional space. The proposed model is solved under the ADMM framework efficiently. Extensive experimental results demonstrate that our model can identify communities more accurately than the baseline methods.

In this paper, we focus on disjoint community detection. Given the fact that real-world networks are sometimes composed of overlapping communities, it is valuable to extend our model to make it suitable for overlapping community detection. The main challenge to achieve this goal is how to preserve the overlapping community structures when embedding nodes into the low-dimensional space.

Acknowledgment

This work was supported by the National Key R&D Program of China (2018YFB1004804), the National Natural Science Foundation of China (61722214, 11801595), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (2016ZT06D211) and the Pearl River S&T Nova Program of Guangzhou (201710010046). Correspondence to Chuan Chen.

References

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

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

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

[4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

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

[6] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 4, p. 54, 2017.

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

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

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

[10] Y. Pei, N. Chakraborty, and K. Sycara, "Nonnegative matrix tri-factorization with graph regularization for community detection in social networks," in *IJCAI*, 2015, pp. 2083–2089.

[11] X. He and P. Niyogi, "Locality preserving projections," in *NIPS*, 2004, pp. 153–160.

[12] S. Chang, G.-J. Qi, C. C. Aggarwal, J. Zhou, M. Wang, and T. S. Huang, "Factorized similarity learning in networks," in *ICDM*. IEEE, 2014, pp. 60–69.

[13] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[14] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering." in *AAAI*, 2016, pp. 1969–1976.

[15] K. Fan, "On a theorem of weyl concerning eigenvalues of linear transformations i," *PNAS*, vol. 35, no. 11, pp. 652–655, 1949.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[17] D. Kim, S. Sra, and I. S. Dhillon, "A non-monotonic method for large-scale non-negative least squares," *Optimization Methods and Software*, vol. 28, no. 5, pp. 1012–1039, 2013.

[18] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l 1-ball for learning in high dimensions," in *ICML*. ACM, 2008, pp. 272–279.

[19] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Mathematical Programming*, vol. 142, no. 1-2, pp. 397–434, 2013.

[20] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[21] L. Yang, X. Cao, D. Jin, X. Wang, and D. Meng, "A unified semi-supervised community detection framework using latent space graph regularization," *IEEE transactions on cybernetics*, vol. 45, no. 11, pp. 2585–2598, 2015.

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

[23] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding." in *AAAI*, 2017, pp. 203–209.