Statistica Sinica **32** (2022), 1-22
doi:https://doi.org/10.5705/ss.202020.0094

COMMUNITY DETECTION IN SPARSE NETWORKS USING THE SYMMETRIZED LAPLACIAN INVERSE MATRIX (SLIM)

Bing-Yi Jing[1], Ting Li[2], Ningchen Ying[1] and Xianshi Yu[1]

[1]*Hong Kong University of Science and Technology and*
[2]*Hong Kong Polytechnic University*

Abstract: There is increasing interest in the study of community detection for sparse networks. Here, we propose a new method for detecting communities in sparse networks that uses the symmetrized Laplacian inverse matrix (SLIM) to measure the closeness between nodes. The idea comes from the first hitting time in random walks, and has a nice interpretation in diffusion maps. Community membership is acquired by applying the spectral method to the SLIM. The SLIM outperforms state-of-art methods in many real data sets and simulations. It is also robust to the choice of tuning parameter, in contrast to spectral clustering with regularization. Theoretical analyses show that in sparse scenarios generated by stochastic block model, the SLIM ensures the same order of misclassification rate in E(degree) as that of regularized spectral clustering.

Key words and phrases: Community, Random walk, Sparse network, Spectral method, Stochastic block model.

1. Introduction

Early research on network community detection focused mostly on dense networks. Many standard approaches are consistent when the networks are sufficiently dense. Their performance on sparse networks is usually unsatisfactory, owing to the intrinsic difficulties caused by sparsity. To illustrate the differences between sparse and dense networks, we plot a stochastic block model (SBM) with different expected degrees in Figure 1. Clearly, a very sparse network comprises many tree-shaped disconnected components. As the network becomes slightly denser ($E(\text{degree}) \geq 1$), a giant connected component (subgraph) emerges. When the network becomes even denser, rings start to appear, and the number of leaf nodes decreases.

Sparse networks are ubiquitous in the real world, for example in social networks and gene co-expression networks. In such cases, the number of objects can

Corresponding author: Bing-Yi Jing, Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. E-mail: majing@ust.hk.

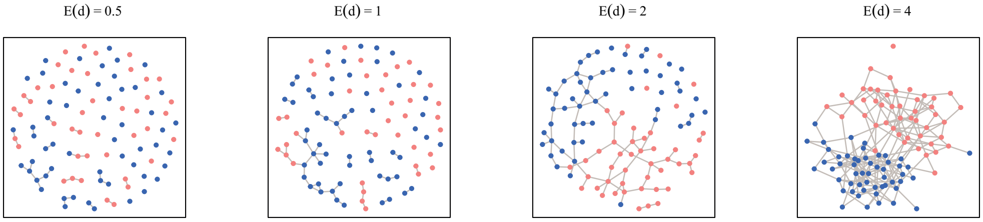


Figure 1. Random networks generated from a SBM: nodes are colored according to true community labels; from left to right, E(degree) grows from 0.5 to 4; for the description of the SBM, please refer to section 3.1; the network generation process is described in section 4.1; here we apply $n = 100$, $K = 2$, $\beta = 0.1$, $\rho = 0$, and $\pi = (1/2, 1/2)$.

reach hundreds of thousands, but the edges are not so easily observable. Methods that are consistent for dense networks often fail in this case, empirically and theoretically. Previous theoretical studies have usually considered the scenario when $E(\text{degree}) = \Omega(\log n)$ (see, e.g., Rohe, Chatterjee and Yu (2011); Lei and Rinaldo (2015); Hajek, Wu and Xu (2016)), where n and *degree* are, respectively, the number of nodes and the number of connections of a single node. However, the scenarios in which $E(\text{degree}) \to \infty$ and $E(\text{degree}) = o(\log n)$ are also of practical interest.

The problem of sparse network community detection is receiving increasing attention. Mossel, Neeman and Sly (2012, 2018) proved the theoretical boundary of obtaining a community estimate that is better than a random guess, where E(degree) is allowed to be as small as $O(1)$. Krzakala et al. (2013) proposed partitioning a sparse network by decomposing its non-backtracking matrix of directed edges. Amini et al. (2013) presented a pseudo-likelihood method and used a regularized version of spectral clustering to generate the initial estimate for the iteration. Bhattacharyya and Bickel (2014) computed the distance matrix to enhance the information in the network. Later, Joseph and Yu (2016) further studied the effect of the regularization method used by Amini et al. (2013). Gao et al. (2017) proved that normalized spectral clustering with regularization can achieve consistency in sparse scenarios. Other recent works on sparse network community detection include, among others, Massoulié (2014) and Chin, Rao and Vu (2015).

In this paper, we propose an alternative method for partitioning sparse networks using the symmetrized Laplacian inverse matrix (SLIM). The SLIM describes the closeness between each pair of nodes. It depicts the indirect connections between nodes by considering a random walk on the network. The SLIM is,

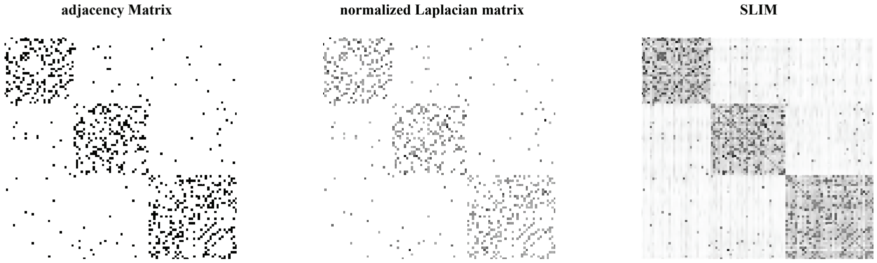


Figure 2. Plot of the adjacency matrix, normalized Laplacian matrix, and the SLIM: We generate a network from an SBM with 100 nodes and three communities. We record the adjacency matrix and compute the normalized Laplacian matrix and the SLIM. We plot these three matrices in grayscale. Within each matrix, the colors *white* and *black* correspond to its smallest and largest entries, respectively; the rows and columns of the matrices have been arranged according to the community structure; the parameters for the SLIM are $\gamma = 0.25$ and $\tau = 0$.

in fact, an approximation of the matrix of exponentially transformed first hitting times, which can also be interpreted as a diffusion map; see Section 2 for details.

There are many intuitive reasons for formulating the SLIM:

- The SLIM is no longer sparse, and it brings out the matrix blocks corresponding to the communities within a network. It works for both sparse and nonsparse networks.
- Because random walks are easily trapped in a community, the first hitting time should be a suitable tool to fulfill the task of community detection.
- The exponential transformation enables the SLIM to emphasize information in the local area and makes the matrix stable.

For illustration, in Figure 2, we plot the SLIM of a network generated from an SBM, defined in section 3. The adjacency matrix and normalized Laplacian matrix are also plotted for comparison. To ease interpretation, the nodes have been ordered according to group indices. The three diagonal blocks of the matrices reflect the intimacy of nodes in the same community. The SLIM fills the zero entries in the adjacency matrix with a positive description of closeness. More importantly, those entries within a community are assigned larger closenesses. A clear contrast is observed at the boundaries of each diagonal block of the SLIM.

We apply the spectral method to the SLIM to estimate the community structures. Here, we theoretically prove the asymptotic consistency of the proposed method under the SBM framework, in both dense ($E(\text{degree}) = \Omega(\log n)$) and

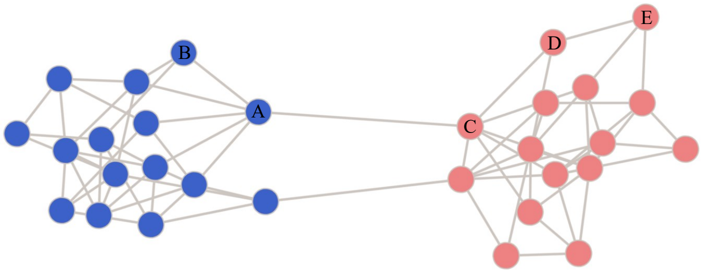


Figure 3. Motivating the SLIM from the first hitting time.

sparse ($E(\text{degree}) \to \infty$ and $E(\text{degree}) = o(\log n)$) scenarios. Specifically, in the sparse scenario, a regularization step is needed in the algorithm to allow the proof of consistency. To the best of our knowledge, for both scenarios, the consistency rate of the SLIM reaches the best among those of all methods realized by applying spectral clustering to a certain matrix. Empirically, we demonstrate that our method is superior to other methods in many settings, for both simulated and real networks. It is always (among) the best, and it is robust in the selection of the regularization parameter.

The remainder of the paper is organized as follows. Section 2 introduces the formulation and the algorithms of the SLIM. Section 3 proves the consistency of the method. In section 4, we demonstrate the performance of the SLIM using numerical experiments. We conclude the paper in section 5. Section S2 in the online Supplementary Material presents detailed proofs of our theorems.

2. Methodology

2.1. Motivation

Let A denote the adjacency matrix, consisting of zeros and ones. For sparse networks, A contains many zero entries, and there is a lack of information on the closeness between nodes. It is desirable to obtain a new matrix that can better depict the closeness between each pair of nodes. We motivate our methodology from two different, but somewhat related angles.

2.1.1. Derivation of the SLIM from the first hitting time in a random walk

Our method is motivated by the first hitting time of random walks. Consider a random walk on a network: starting from each node, one of its edges is chosen

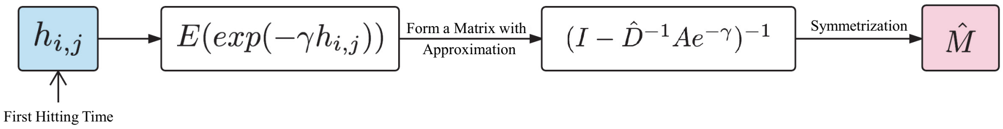

$h_{i,j}$
First Hitting Time
$E(exp(-\gamma h_{i,j}))$
Form a Matrix with Approximation
$(I - \hat{D}^{-1}Ae^{-\gamma})^{-1}$
Symmetrization
$\hat{M}$

Figure 4. The idea behind the formulation of the SLIM $\hat{M}$.

with equal probability. For example, in Figure 3, the first hitting time from C to B is larger than that from C to E, despite their graph distances both being 2.

Let $h_{i,j}$ denote the first hitting time from node i to node j. Then, $E(\exp(-\gamma h_{i,j}))$ is a good local similarity measure between the two nodes, where the exponential transformation emphasizes the local information by down-weighting the large first hitting time. However, $E(\exp(-\gamma h_{i,j}))$ is very difficult to calculate. Therefore we approximate it by

$$H = \sum_{k=1}^{\infty} \exp(-\gamma k)(\hat{D}^{-1}A)^k = \sum_{k=1}^{\infty} \alpha^k (\hat{D}^{-1}A)^k,$$

where $\alpha = e^{-\gamma}$, A is the adjacency matrix, $\hat{D}$ is the diagonal matrix of degrees, and $\hat{D}^{-1}A$ is the transition matrix of a random walk on the network. In this approximation, instead of counting only the first hitting time, we count all hitting times. Because $\exp(-\gamma k)$ is very small when k is large, the approximation is reasonable.

It is easy to see that $H = (I - \alpha\hat{D}^{-1}A)^{-1} - I$. We denote the inverse of the Laplacian matrix

$$(I - \alpha\hat{D}^{-1}A)^{-1}$$

by $\hat{W}$, which has the same eigenvectors as H. From $\hat{W}$, we can define the symmetrized Laplacian inverse matrix (SLIM):

$$\hat{M} = \frac{\hat{W} + \hat{W}^*}{2}.$$

Figure 4 depicts the idea behind its formulation, and Figure 2 shows the plot of $\hat{M} - I$. Under the SBM, $E(\hat{M}) - I$ is a block matrix.

2.1.2. Derivation of the SLIM from the diffusion map

The formulation of $\hat{W}$ can also be motivated by the idea of a diffusion map. Note that $\hat{W} = H + I$ is a power series of the transition matrix $\hat{D}^{-1}A$. Coifman and Lafon (2006) interpreted $(\hat{D}^{-1}A)^k$ as an integration of the local geometry of

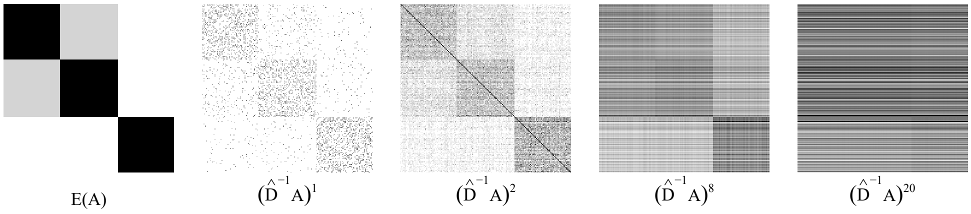


Figure 5. Plots of different powers of the transition matrix: We generate a network with 200 nodes and three communities from an SBM, and compute $(\hat{D}^{-1}A)^k$ with different ks; the left-most plot of $E(A)$ shows the model setting; the rows and columns of the matrices are ordered according to the community indices.

the system, where k is the scale of integration. As k increases, small cliques start to merge and, eventually, all nodes merge into a single group. Under this idea, $\hat{W}$ is a weighted summation of different scales of representations of the network structure.

Figure 5 shows plots of $(\hat{D}^{-1}A)^k$. For $k = 1$, it is difficult to observe the community structure. With increasing k, the structure becomes clearer. As k becomes even larger, the boundaries between the communities become blurred. Eventually, $(\hat{D}^{-1}A)^k$ no longer contains any block structure, and all nodes merge into one giant community. It is reasonable to adopt a decreasing weight of $e^{-k\gamma}$ for this summation, because the matrix power contains very little group information as k increases.

In practice, it suffices to replace $\hat{W}$ with a finite number m of matrix powers, resulting in

$$\hat{W}_m = \sum_{k=1}^{m} \alpha^k (\hat{D}^{-1}A)^k, \quad \text{where} \quad \alpha = e^{-\gamma}. \tag{2.1}$$

This is useful for large networks, where the calculation of a matrix inverse is very time consuming. We include the performance of $\hat{W}_m$ in section 4. Furthermore, in appendix 4.2.3, we specially examine the performance of this approach. The simulation results indicate that $\hat{W}_m$ behaves similarly to $\hat{W}$ even for small m's.

2.2. Algorithm for community detection using the SLIM

The estimate of community indices is obtained by decomposing $\hat{M}$.
Several remarks are in order.

- The method using A to compute the SLIM is called the *SLIM method*, and the method using A_τ is referred to as *SLIM with regularization*, denoted by

Define the similarity matrix using the SLIM:

1. Calculate the inverse Laplacian matrix $\hat{W} = (I - \alpha\hat{D}^{-1}A)^{-1}$, where $\alpha = e^{-\gamma}$ and $\hat{D} = diag(A \times \mathbf{1}_{n\times 1})$.
2. Calculate $\hat{M} = (\hat{W} + \hat{W}^*)/2$.
3. Force the diagonal entries of $\hat{M}$ to zero.

*. For the SLIM with regularization, replace the A and $\hat{D}$ in step 1 with A_τ and $\hat{D}_\tau = diag(A_\tau \times \mathbf{1}_{n\times 1})$ respectively, where $A_\tau = A + (\tau/n)\mathbf{1}\mathbf{1}^T$.

Perform spectral clustering:

4. Perform spectral decomposition on $\hat{M}$, and find the first k eigenvectors. Here, the eigenvectors are ordered according to the eigenvalues in decreasing order.
5. Align these eigenvectors as columns to form an $n \times k$ matrix X.
6. Consider the rows of X as positions of the nodes, and apply clustering methods to obtain the community label of each node.

SLIM$_\tau$. Furthermore, we use *SLIM methods* to refer to all methods that apply the SLIM (matrix).

- For sparse networks, a small perturbation τ is added to A to make $\hat{D}^{-1}$ stable, following Amini et al. (2013); Joseph and Yu (2016); Gao et al. (2017). The perturbation guarantees the theoretical consistency of sparse networks. In practice, though, the SLIM method works well enough; see section 4.

- The diagonal entries of $\hat{M}$ are set to a constant before the decomposition, which is found to work well in practice. This step is not included in the theoretical analysis though.

- Steps 4 to 6 are standard steps for spectral clustering, and the clustering method in the last step can use the k-means, Gaussian mixture model based EM Algorithm and so on. The k-means is often used in prior studies. In our experiments, we find that partitioning around medoids (PAM) Kaufman and Rousseeuw (1990) works consistently well for the SLIM methods.

- The computational complexity of the SLIM methods is $O(\max(n^2, C_{\text{inverse}}, C_{\text{svd}-\text{k}}, C_{\text{kmeans}}))$. Here, C_{inverse}, $C_{\text{svd}-\text{k}}$, and C_{kmeans} are the computational complexities of the matrix inversion, k-top SVD for the $n \times n$ matrices, and k-means with n k-dimensional observations, respectively, which depend on the specific algorithms applied. Taking $C_{\text{inverse}} = O(n^3)$, $C_{\text{svd}-\text{k}} = O(n^3)$, and $C_{\text{kmeans}} = O(Ik^2n)$, the computational complexity of the SLIM methods is then $O(\max(n^3, Ik^2n))$. Here I is the number of iterations in the k-means.

- When handling a large network, we suggest approximating $\hat{W}$ by $\sum_{k=1}^{m}\alpha^k(\hat{D}^{-1}A)^k$, rather than calculating $\hat{W}$ explicitly. This approximation approach (SLIMappro) is examined using numerical experiments in section 4.

2.3. Choice of tuning parameters

We need to set two tuning parameters, τ and γ.

- We recommend choosing $\tau = c\tilde{d}$, for a small constant $c > 1$, where $\tilde{d}$ is the observed average degree of the nodes. In section 4, we show that the SLIM is not very sensitive to the amount of perturbation τ. Empirically, we find $\tau = 0.1\tilde{d}$ to be a consistently good choice.

- We recommend setting $\gamma = 0.25$. This has been found to be a good choice for many different scenarios, including the SBM and the degree-corrected SBM. This choice of γ is applied in all of the numerical studies discussed in this paper.

Overall, our recommendation is to use the SLIM$_\tau$ with τ being 0.1 times the observed average degree of nodes and $\gamma = 0.25$, by default. When n is large, we recommend applying the SLIMappro approach to accelerate the computation.

3. Main Results

In this section, we show that the SLIM is consistent in the sense that the error rate approaches zero asymptotically in the context of an SBM.

3.1. Model assumptions and notation

The SBM assumes that in generating the edges, there is a $K \times K$ symmetric matrix B that guides the process, where K is the number of communities. The edges between pairs of nodes are generated independently, and node i and node j are connected with probability $b_{g_i g_j}$, where g_i is the group index of node i. We store the group indices in a matrix $\Theta \in \mathbb{F}_{n,K}$, with $\mathbb{F}_{n,K}$ the collection of all $n \times K$ matrices, such that each row is composed of a single one and $(K-1)$ zeros. Here Θ is called a membership matrix, and $\Theta_{i,g_i} = 1$.

An observed network is represented by its adjacency matrix $A_{n\times n}$, a 0-1 symmetric matrix. Let $P = \Theta B \Theta^T$. Then, $A_{i,j}(i < j)$ follows an independent Bernoulli distribution, with $p = p_{i,j} = b_{g_i g_j}$. The edge generating process is illustrated as follows:

$$P = \begin{pmatrix} b_{1,1}\mathbf{1}_{n_1\times n_1} & b_{1,2}\mathbf{1}_{n_1\times n_2} & \cdots & b_{1,K}\mathbf{1}_{n_1\times n_K} \\ b_{2,1}\mathbf{1}_{n_2\times n_1} & b_{2,2}\mathbf{1}_{n_2\times n_2} & \cdots & b_{2,K}\mathbf{1}_{n_2\times n_K} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K,1}\mathbf{1}_{n_K\times n_1} & b_{K,2}\mathbf{1}_{n_K\times n_2} & \cdots & b_{K,K}\mathbf{1}_{n_K\times n_K} \end{pmatrix}.$$

For example, when $K = 2$,

$$P = \begin{pmatrix} 0.8 & 0.8 & 0.8 & 0.2 & 0.2 \\ 0.8 & 0.8 & 0.8 & 0.2 & 0.2 \\ 0.8 & 0.8 & 0.8 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.6 & 0.6 \\ 0.2 & 0.2 & 0.2 & 0.6 & 0.6 \end{pmatrix} \Rightarrow A = \begin{pmatrix} 1 & 1 & 1 & & \\ 1 & 1 & & & 1 \\ 1 & & 1 & & \\ & & & 1 & \\ & 1 & & & 1 \end{pmatrix}.$$

Here, the nodes have been ordered according to their group memberships. Clearly, $E(A) = P$ and the task is to recover the block structure of P from A, whose nodes are unordered.

We now define the error rate to evaluate the performance of the community detection. Let $\hat{\Theta} \in \mathbb{F}_{n,K}$ be the estimated membership matrix. We consider the overall proportion of misclassified nodes,

$$L(\hat{\Theta}, \Theta) = n^{-1} \min_{Q \in E_K} \|\hat{\Theta}Q - \Theta\|_0, \tag{3.1}$$

where E_K is the set of all $K \times K$ permutation matrices. This value is sometimes called the "misclassification rate."

Define

$$M = \frac{1}{2}((I - \alpha D^{-1}P)^{-1} + ((I - \alpha D^{-1}P)^{-1})^T), \text{ where } D = diag(P\mathbf{1}) \text{ and } \alpha = e^{-\gamma}.$$

We define M_τ similarly by replacing P and D above with $P_\tau = P + \frac{\tau}{n}\mathbf{1}\mathbf{1}^{\mathbf{T}}$ and $D_\tau = diag(P_\tau \mathbf{1})$, respectively. Under the SBM, M and M_τ have a block structure corresponding to the group structure in the network. Here, $\hat{M}$ and $\hat{M}_\tau$ are close to M and M_τ, respectively, with high probability.

3.2. Main result

We now show the consistency of the SLIM methods under different scenarios.

3.2.1. Consistency of the SLIM with regularization for sparse networks

For sparse networks, we eliminate the nodes with degree larger than Cd, where C is a sufficiently large constant, and d is the expectation of the average degree, which can be estimated by the mean of the observed degrees. This step is only for the technical proof. It makes little difference in a real application if we take C to be sufficiently large.

Theorem 1. *Let $\hat{\Theta}_\tau$ be the membership matrix using the SLIM with regularization with $\tau \in [C_1 d, C_2 d]$, for some large constants $C_1, C_2 > 0$, taking α as any constant in $(0,1)$, and using the k-means as the clustering method. Here, $d= np_{max} + 1$ and $p_{max} = \max_{u \geq v} P_{uv}$. Then, as $n \to \infty$, for any $C' > 0$, there exists a constant $C > 0$, and with probability $1 - n^{-C'}$, we have*

$$L(\hat{\Theta}_\tau, \Theta) \leq C \frac{\log d}{(\gamma_{\tau,K} - \gamma_{\tau,K+1})^2 d},$$

if $\log d/((\gamma_{\tau,K} - \gamma_{\tau,K+1})^2 d) \leq \varepsilon$, for some small $\varepsilon \in (0,1)$, here $\gamma_{\tau,K}$ is the Kth-largest eigenvalue of M_τ.

The proof of Theorem 1 is given in appendix S2. Theorem 1 reveals that the upper bound of the misclassification rate is negatively correlated with the eigen gap $(\gamma_{\tau,K} - \gamma_{\tau,K+1})$ of M_τ and the average degree d. Moreover, the eigen gap of M_τ is determined by the values of the SBM parameters and the choice of τ. The interpretation of Theorem 1 becomes much easier in the following simple case.

Condition 1. *$B = (a/n - b/n) I_K + (b/n) \mathbf{1}_K \mathbf{1}_K^T$ and $n_1 = n_2 = \cdots = n_K = n/K$. Here, I_K is the $K \times K$ identity matrix, and $a > b > 0$.*

Corollary 1. *Assume the conditions in Theorem 1 and Condition 1 hold. Then, as $n \to \infty$, for any $C' > 0$, there exists a constant $C > 0$, such that, with probability $1 - n^{-C'}$,*

$$L(\hat{\Theta}_\tau, \Theta) \leq C \left(\frac{\tau}{a-b} \right)^2 \frac{\log a}{a}.$$

Remark 1. Because a/n and b/n indicate the possibility of an edge existing between a pair of nodes from the same group and from two different groups, respectively, $a-b$ in Corollary 1 reflects the strength of the signal. From Corollary 1, recalling that $\tau \in [C_1 d, C_2 d]$, τ cannot be too large, and $a - b$ cannot be too small. Moreover, as $a \to \infty$, the error rate goes to zero, as long as $\sqrt{a \log a}/(a-b)$ goes to zero.

3.2.2. Consistency of the SLIM method for dense networks

The results for dense networks are stated below. We omit the details of the proofs because they are similar to those in Lei and Rinaldo (2015).

Theorem 2. *Let $\hat{\Theta}$ be the membership matrix obtained using the SLIM, with α as any constant in $(0,1)$, and using the k-means as the clustering method. Then,* *as $n \to \infty$, for any $C' > 0$, there exists a constant $C > 0$, and with probability $1 - n^{-C'}$, we have*

$$L(\hat{\Theta}, \Theta) \leq C \frac{\log d}{(\gamma_K - \gamma_{K+1})^2 d},$$

if $\log d/((\gamma_K - \gamma_{K+1})^2 d) \leq \varepsilon$, for relatively small ε in $(0,1)$, and $d_{min} = \Omega(\log n)$; here γ_K is the Kth-largest eigenvalue of M.

Remark 2. To the best of our knowledge, for the SLIM (SLIM$_\tau$) method, the consistency rates in both dense and sparse regimes are among the best, considering all one-step spectral algorithms. Its rate is the same as that achieved by *normalized spectral clustering with regularization* (Theorem 4 in Gao et al. (2017)). Because the SLIM method satisfies the weak consistency condition in Gao et al. (2017), if it is adopted as the initialization method for Algorithm 1 in Gao et al. (2017), the optimal misclassification proportion proposed in Zhang and Zhou (2016) can be achieved. We demonstrate in the next section that the SLIM usually outperforms normalized spectral clustering with regularization empirically. It is much more independent of the choice of the regularization term τ.

4. Numerical Results

We investigate the performance of the SLIM using simulated and real networks. In section 2, we introduced the SLIM method and the SLIM with regularization (SLIM$_\tau$). In this section, we examine the performance of both methods, as well as the approximation approach (SLIMappro) introduced in (2.1). The following methods/algorithms are adopted for comparison purposes:

- normalized spectral clustering (SC),
- normalized spectral clustering with regularization (SC$_\tau$) Amini et al. (2013); Joseph and Yu (2016),
- spectral clustering on ratios-of-eigenvectors (SCORE) Jin (2015),
- spectral algorithms based on a non-backtracking walk (NB) Krzakala et al. (2013),
- a pseudo-likelihood algorithm (PL) Amini et al. (2013), and
- a pseudo-likelihood conditional on node degrees (CPL) Amini et al. (2013).

Among these methods, SC represents the traditional approaches. In addition, SC$_\tau$, NB, PL, and CPL represent the state of the art for sparse community detection, and SCORE is the benchmark for the degree-corrected SBM (DCSBM).

The first four methods and the SLIM method are spectral methods. For all of them, we applied k-means as the clustering method in the last step. PL and CPL adopt the expectation-maximization (EM) algorithm. Following the suggestions in Amini et al. (2013), we applied SC_τ to initialize the iterations of PL and CPL.

The numerical results suggest that the SLIM (even without regularization) successfully addresses the sparsity issue in network community detection. Its performance is robust with respect to the regularization parameter τ, in contrast to SC_τ. Furthermore, simulation results in various model settings show that the proposed methods (SLIM, SLIM_τ, and SLIMappro) are competitive with cutting-edge methods. Furthermore, the SLIM works well, not only in the standard SBM setting, but also for the DCSBM. Moreover, the SLIM method maintains the best accuracy in all three of our real data experiments.

In section 4.1, we introduce the network generation scheme implemented in our simulations, criteria adopted as performance measures, and the default parameters of the methods. The simulation results follow. In section 4.2, we investigate the SLIM method and find the scenarios it specializes in. The performance of SLIM_τ and SLIMappro, with varying parameters, is also studied here. Section 4.3 compares the SLIM methods and the methods listed above. The results of the real-data analysis are given in section 4.4.

4.1. Network generation scheme, performance measure, and default parameters

Throughout the simulation studies, we use an SBM (DCSBM) to generate networks with 1,200 nodes and three communities. We follow the simulation scheme in Amini et al. (2013). The community labels of the nodes are the outcomes of independent multinomial draws, with $\pi = (\pi_1, \pi_2, \pi_3)$. Conditioning on these labels, the edges are generated as independent Bernoulli variables, with $p = B_{g_i g_j}$, while under the DCSBM, $p = \theta_i \theta_j B_{g_i g_j}$. We use θ_i to represent the popularity of node i, and θ_i are drawn independently, with $P(\theta = 0.2) = \rho$ and $P(\theta = 1) = 1 - \rho$. We consider two settings, namely, $\rho = 0$ and $\rho = 0.9$, which correspond to the standard SBM and the DCSBM, respectively.

The block probability matrix B is controlled by two parameters: the overall edge density λ, and the "out-in-ratio" β. Here, λ is $E(degree)$, and λ ranges from 2 to 10 in our simulations, where a small λ indicates a sparse networks. In addition, β controls the ratio between the inter- and intra-community connection probabilities, and is set between 0.02 and 0.2. The generation of B consists two steps:

1. Generate $B^{(0)}$, whose diagonal and off-diagonal entries are set to one and β, respectively.

2. Obtain B by rescaling $B^{(0)}$, making $E(degree) = \lambda$. Specifically,

$$B = \frac{\lambda}{(n-1)(\pi^T B^{(0)} \pi)(\mathbb{E}\theta)^2} B^{(0)}. \tag{4.1}$$

All simulations discussed below adopt the same network generation scheme as described above. We control the parameters λ, β, ρ, and π to model different settings. Under each parameter setting, we replicate the simulation process and report the average performance of the methods. The number of replications is 100, unless otherwise stated. In order to be coherent with our theories, we adopt the missclassification rate (3.1) as the measure of performance. Note that other performance measures are possible; see Liu, Cheng and Zhang (2019).

Note that throughout the numerical studies, we use $\gamma = 0.25$ for the SLIM methods, as suggested in section 2.3. Furthermore, in sections 4.3 and 4.4, we apply $\tau = 0.1$ for both SC$_\tau$ and the SLIM$_\tau$. This choice was made referring to the experimental results shown in section 4.2.2.

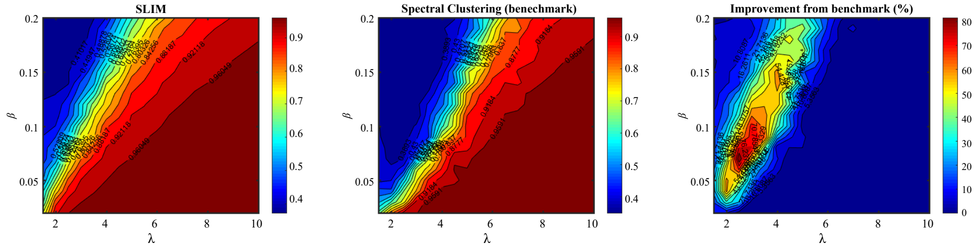
4.2. Performance of the SLIM method, SLIM$_\tau$, and SLIMappro

Before comparing the methods, we examine the experimental properties of the SLIM.

4.2.1. The SLIM addresses the sparse issue

In this section, we compare the accuracy of the SLIM and the SC in the SBM setting, with varying average degree (λ) and "out-in ratio"(β). Here, we apply the SLIM method ($\tau = 0$). Note that SLIM$_\tau$ has the potential to improve the SLIM method, with a proper choice of τ. In the simulations, we fix $\pi = (1/3, 1/3, 1/3)$ and $\rho = 0$. Figure 6 shows the performance of the SLIM method and the SC, where the color indicates average accuracy.

The figures show the deficiency boundary of the SLIM method and the SC. When the "out-in ratio" is large, or when the expected degree is small, the accuracy is low. This observation coincides with the numerical results reported in previous works, and validates our statement that the sparse scenario is difficult and needs special treatment. The SLIM method successfully pushes the deficiency boundary toward the upper-left corner (see the comparison between Figure 6b and 6a). Figure 6c presents the improvement of the SLIM method over the SC. It appears in the sparse area, at the deficiency boundary of the SC. This observation



(a) ACCURACY of the SLIM (no regularization.)

(b) ACCURACY of the SC.

(c) Difference between the SLIM and SC.

Figure 6. Comparison between the SC and SLIM (without regularization): Networks are simulated from an SBM with $n = 1{,}200$, $K = 3$, $\rho = 0$, $\pi = (1/3, 1/3, 1/3)$, and varying λ and β; (c) shows the result of subtracting (b) from (a).

justifies our motivation and intuition for presenting the SLIM. Furthermore, it suggests that the perturbation τ may not be necessary for sparse scenarios.

4.2.2. The effect of τ to SLIM$_\tau$

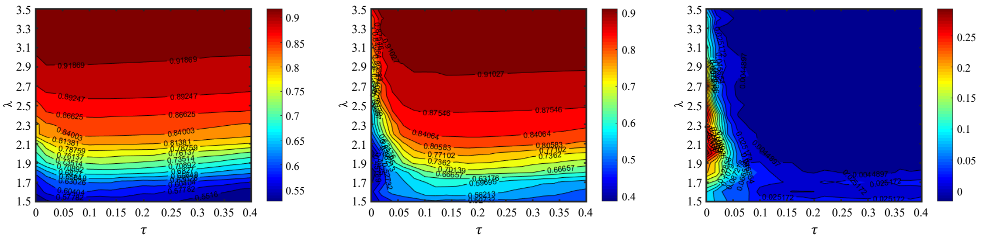
The regularization parameter τ is introduced in the SLIM$_\tau$ to ensure theoretical consistency in sparse scenarios. This form of regularization was first introduced in SC$_\tau$ by Amini et al. (2013) to improve the performance of the SC in sparse scenarios, and has proven effective, both empirically and theoreticallyAmini et al. (2013); Gao et al. (2017); Joseph and Yu (2016). However, to the best of our knowledge, there is no practical criterion for choosing the best τ. Previous analysis has only suggested that τ should have the same order as the observed average degree. In figure 7, we show the effect of τ for SLIM$_\tau$ and SC$_\tau$. For both methods, the performance at $\tau = 0$ is also included. This corresponds to the SLIM method and SC, respectively. For ease of presentation, we only specify the ratio between τ and the observed average degree. For example, $\tau = 2$ actually means $\tau = 2\sum \hat{d}_i/n$. The simulations are carried out under an SBM with $\pi = (1/3, 1/3, 1/3)$, $\rho = 0$, and varying λ.

From Figures 7, we have the following observations:

- Under the adopted SBM setting, the best τ for both methods is about $0.1\sum \hat{d}_i/n$.

- For both the SLIM$_\tau$ and SC$_\tau$, either a τ that is too small or too large results in suboptimal performance. However, as τ decreases from the optimal choice, the accuracy of SC$_\tau$ decreases abruptly, while the performance of SLIM$_\tau$ is much more stable (see the comparison between Figure 7a and 6b).

- Figure 7c shows the improvement of SLIM$_\tau$ over SC$_\tau$. When the same



(a) ACCURACY of SLIM_τ. (b) ACCURACY of SC_τ. (c) (a) − (b).

Figure 7. The effect of τ in SLIM_τ and SC_τ: Here, $n = 1{,}200$, $K = 3$, $\beta = 0.05$, $\rho = 0$, $\pi = (1/3, 1/3, 1/3)$, and λ varies. (c) shows the result of subtracting (b) from (a); the results at $\tau = 0$ show the performance of the SLIM method and SC respectively.

τ is applied in both methods, SLIM_τ improves SC_τ the most at small τ. Moreover, for all τ, we observe a slight advantage for SLIM_τ in sparse areas.

Note that SLIM_τ is less dependent on the choice of τ than SC_τ is. This property is desirable. Unlike for SC_τ, in real applications, a small τ is always a safe choice for SLIM_τ. Furthermore, the simulation results above show that the regularization term τ might not be necessary. However, we are not able to prove this at present. We give real-data experiments for τ in section 4.4.1.

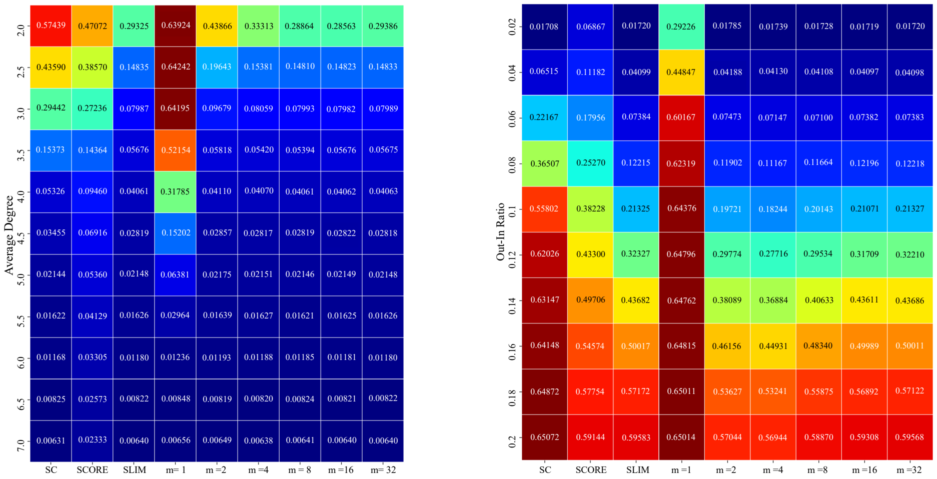
4.2.3. The performance of SLIMappro

In section 2.1.2, we suggested an approximation approach for the SLIM to accelerate its computation; when n is large,

$$\hat{W}_m = \sum_{k=1}^{m} \alpha^k (\hat{D}^{-1} A)^k, \text{ where } \alpha = e^{-\gamma}.$$

Here, we experimentally examine the performance of this approach, with m ranging from 1 to 32. We repeatedly generate networks from an SBM with $\rho = 0$ and $\pi = (1/3, 1/3, 1/3)$. We vary λ and β, and report the average misclassification rate obtained from $\hat{W}_m$ in Figure 8. The performance of the SC, SCORE, and SLIM method are also shown, for comparison. Specifically, for both the SLIM method and the approximation approach, we used $\gamma = 0.25$ and $\tau = 0$.

In the figure above, the average misclassification rate of each setting is represented by color, where a blue color is better. We observe that $\hat{W}_m$, with m as small as 2, performs satisfactorily. Furthermore, in the simulated settings, this approximation approach behaves very similarly to the SLIM method. In the case of a low "out-in" ratio, it is even better. These results suggest that, in cases of large networks, we can safely use this approximation approach with a small



Average Degree	SC	SCORE	SLIM	m= 1	m =2	m =4	m = 8	m =16	m= 32
2.0	0.57439	0.47072	0.29325	0.63924	0.43866	0.33313	0.28864	0.28563	0.29386
2.5	0.43590	0.38570	0.14835	0.64242	0.19643	0.15381	0.14810	0.14823	0.14833
3.0	0.29442	0.27236	0.07987	0.64195	0.09679	0.08059	0.07993	0.07982	0.07989
3.5	0.15373	0.14364	0.05676	0.52154	0.05818	0.05420	0.05394	0.05676	0.05675
4.0	0.05326	0.09460	0.04061	0.31785	0.04110	0.04070	0.04061	0.04062	0.04063
4.5	0.03455	0.06916	0.02819	0.15202	0.02857	0.02817	0.02819	0.02822	0.02818
5.0	0.02144	0.05360	0.02148	0.06381	0.02175	0.02151	0.02146	0.02149	0.02148
5.5	0.01622	0.04129	0.01626	0.02964	0.01639	0.01627	0.01621	0.01625	0.01626
6.0	0.01168	0.03305	0.01180	0.01236	0.01193	0.01188	0.01185	0.01181	0.01180
6.5	0.00825	0.02573	0.00822	0.00848	0.00819	0.00820	0.00824	0.00821	0.00822
7.0	0.00631	0.02333	0.00640	0.00656	0.00649	0.00638	0.00641	0.00640	0.00640

(a) MISCLASSIFICATION RATE with varying λ and m.

Out-In Ratio	SC	SCORE	SLIM	m =1	m =2	m =4	m =8	m =16	m =32
0.02	0.01708	0.06867	0.01720	0.29226	0.01785	0.01739	0.01728	0.01719	0.01720
0.04	0.06515	0.11182	0.04099	0.44847	0.04188	0.04130	0.04108	0.04097	0.04098
0.06	0.22167	0.17956	0.07384	0.60167	0.07473	0.07147	0.07100	0.07382	0.07383
0.08	0.36507	0.25270	0.12215	0.62319	0.11902	0.11167	0.11664	0.12196	0.12218
0.1	0.55802	0.38228	0.21325	0.64376	0.19721	0.18244	0.20143	0.21071	0.21327
0.12	0.62026	0.43300	0.32327	0.64796	0.29774	0.27716	0.29534	0.31709	0.32210
0.14	0.63147	0.49706	0.43682	0.64762	0.38089	0.36884	0.40633	0.43611	0.43686
0.16	0.64148	0.54574	0.50017	0.64815	0.46156	0.44931	0.48340	0.49989	0.50011
0.18	0.64872	0.57754	0.57172	0.65011	0.53627	0.53241	0.55875	0.56892	0.57122
0.2	0.65072	0.59144	0.59583	0.65014	0.57044	0.56944	0.58870	0.59308	0.59568

(b) MISCLASSIFICATION RATE with varying β and m.

Figure 8. MISCLASSIFICATION RATE of the approximation approach using $\hat{W}_m$: This approach is tested with different m (from 1 to 32); the last six columns correspond to SLIMappro, with m as indicated below; networks are simulated from an SBM with $n = 1{,}200$, $K = 3$, $\rho = 0$, and $\pi = (1/3, 1/3, 1/3)$; in (a), $\beta = 0.05$; in (b), $\lambda = 3.5$; for each parameter setting, the number of replications is 100.

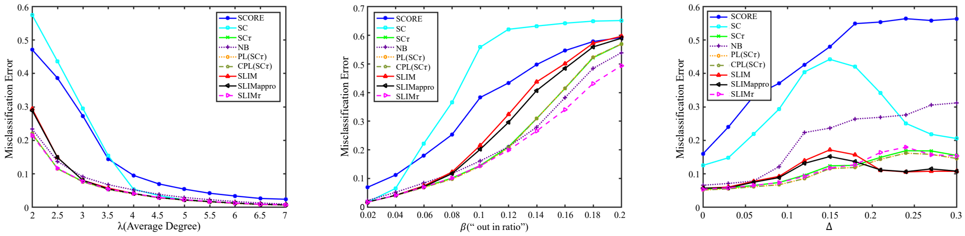
$m \geq 2$. We show more results using this approach, with $m = 8$, in sections 4.3 and 4.4.

4.3. Comparison with other methods

We carry out extensive simulations to compare the SLIM methods with cutting-edge methods introduced for sparse networks in the literature. All three SLIM methods examined above are considered. In the notation of the figures, "SLIM" stands for the SLIM method ($\tau = 0$), "SLIM$_\tau$" is applied with $\tau = 0.1$, and "SLIMappro" is applied with $m = 8$. We implement the same choice of τ for SC$_\tau$. Because the model settings for the simulations in this section are similar to those in Figure 7, we believe the best choice for τ should be close to 0.1. Moreover, for both PL and CPL, we adopt SC$_{\tau=0.1}$ to initialize the iterations.

4.3.1. Standard SBM

We start the simulations under the standard SBM setting ($\rho = 0$). We run three groups of simulations to test the methods, with varying λ, β and π. Specifically, for π, we use $\pi = (1/3 - \Delta, 1/3, 1/3 + \Delta)$, with Δ varying between zero and 0.3. Here, Δ can be interpreted as the degree of imbalance in the



(a) ERROR RATE with varying λ.

(b) ERROR RATE with varying β.

(c) ERROR RATE with varying Δ.

Figure 9. Comparison in standard SBM: For all three figures, $n = 1{,}200$, $K = 3$, and $\rho = 0$; $\beta = 0.05$ in (a) and (c); $\Delta = 0$ in (a) and (b); $\lambda = 3.5$ in (b) and (c).

community size.

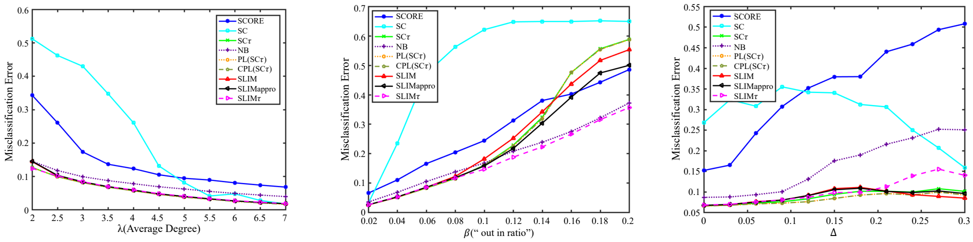
Figure 9a shows the performance of the methods as the networks change from sparse to dense. Figure 9b shows their performance as the "out-in" ratio changes. In general, a larger β means there is a smaller "contrast" in the observed networks, and therefore harder tasks. On the other hand, we also consider varying π, because imbalances in group sizes could be a practical issue in real applications. Figure 9c shows the performance of the tested methods pertaining to this issue.

Overall, the SLIM methods behave satisfactorily. In Figure 9a, SLIM$_\tau$, SC$_\tau$, PL, and CPL have the same and the lowest misclassification rates. Figure 9b shows that SLIM$_\tau$ outperforms the others considerably in the case of "low contrast," that is when the "out-in ratio" is large. When the community sizes are nonhomogeneous, the misclassification rate of SLIM$_\tau$ is still close to the lowest rate (PL and CPL). Moreover, the SLIM method ($\tau = 0$) and SLIMappro perform acceptably as well, with large improvements over the SC in all settings. Note that SLIMappro behaves comparably with the SLIM method.

4.3.2. DCSBM

We repeat the simulations above in DCSBM. The only difference here is that, in the generation of the networks, 10% of the nodes are hubs with high popularity. The results are shown in Figure 10.

We obtain similar observations from this set of simulations to those in section 4.3.1. The SLIM methods are still competitive. Note that all methods designed for sparse networks work well in the simulated DCSBM settings. Though traditional methods tend to deteriorate when the degrees of nodes are heterogeneous Jin (2015), the methods for sparse networks largely improve the performance of the SC. They even outperform SCORE in most of the applied settings.



(a) ERROR RATE with varying λ.

(b) ERROR RATE with varying β.

(c) ERROR RATE with varying Δ.

Figure 10. Comparison in DCSBM: Here $\rho = 0.9$; n, K, β, λ, and Δ are as in Figure 9.

Table 1. Data description

	n (# of nodes)	K (# of communities)	average degree
Politic blogs	1,222	2	27.36
Politic books	105	3	8.40
College football	115	12	7.67

4.4. Real-Data Analysis

In this section, we examine the performance of the SLIM using real network data. Those methods considered in the simulations above are applied here as well. Three commonly studied data sets are used, which can be downloaded at `https://github.com/yningc/SLIM/tree/master/SLIM_MATLAB`.

Political blog network (Pbl) Adamic and Glance (2005) is regarded as a typical degree-corrected network Jin (2015). These data were collected immediately after the 2004 US presidential election. Pairs of blogs are connected if there is a hyperlink between them. The giant component contains 1,222 blogs and 16,714 edges, where each blog is manually labeled as either liberal or conservative. The belief that blogs with similar political attitudes tend to connect makes this network ideal for a network community study. Many researchers have tested their methods on this data set to see how close their community detection results are to the manual labels.

Political books network (Pbk) contains 105 nodes and 441 edges. Nodes are US political books sold by Amazon.com, and edges represent the occurrence of co-purchasing. Nodes have been manually partitioned into three groups, namely, liberal, neutral, and conservative. These partitions are adopted as true community labels in the measure of accuracy.

Table 2. MISCLASSIFICATION RATE (%) of methods: We considered the same methods here as in section 4.3; for a detailed description of the methods and the parameters used, please refer to the beginning of section 4.3; here, $\tau = 0.1$ means $\tau = 0.1 \sum \hat{d}_i/n$; the lowest misclassification rate in each row is presented in bold face.

	SLIM			SCORE	SC	SC_τ	NB	PL	CPL
data	$\tau = 0$	appro	$\tau = 0.1$						
Pbl	**4.26**	4.34	5.16	4.75	49.02	18.82	5.32	4.75	4.99
Pbk	16.19	16.19	**15.23**	24.76	16.19	16.19	18.10	17.14	17.14
Cfb	**7.83**	**7.83**	**7.83**	10.43	**7.83**	10.43	15.65	11.30	11.30

College football network (Cfb) is derived from the schedule of Division I games for the 2000 season in the United States Girvan and Newman (2002). It has 115 nodes, representing the football teams, and the 441 edges, indicating regular-season games between pairs of teams. This network has a natural community structure inherited from the formation of 12 conferences. Each contains 8–12 teams. Games were more frequent within a conference than between members of different conferences. On average, each team played about seven intra-conference games and four inter-conference games. This fact makes it possible to infer the conference membership of teams from the network structure.

Table 1 summarizes the data and Table 2 reports the performance of the considered methods. The set of methods examined here is the same as that in section 4.3.

The SLIM performs very competitively, reporting the lowest misclassification rate in all three networks.

Note too that, for the political blogs data, the SLIM method is better than SLIM_τ. This indicates that, for this particular data, a τ smaller than $0.1 \sum \hat{d}_i/n$ should be used. Although the network size of this data is similar to that of simulation shown in Figure 7, the best τ is quite different. This observation validates our statement that, in applications, the selection of τ for SLIM_τ and SC_τ could be tricky. Therefore, the robustness of the SLIM with respect to τ is crucial. We perform additional experiments for the choice of τ below.

4.4.1. Apply SLIM_τ and SC_τ to the political blogs network with different τ

Figure 11 shows the misclassification rate of SLIM_τ and SC_τ with varying τ. Specifically, for each method, we try both k-means and partitioning around medoids (PAM) Kaufman and Rousseeuw (1990) as clustering methods in the last step.

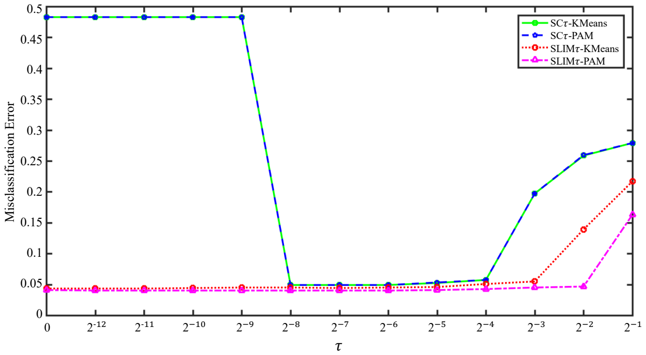


Figure 11. MISCLASSIFICATION RATE of SLIM_τ and SC_τ on the political blogs network; the x-coordinate is the ratio between τ and $\sum \hat{d}_i/n$.

This result indicates that the SLIM gives satisfactory performance, as long as τ is smaller than $0.125 \sum \hat{d}_i/n$, while for SC_τ, one has to use τ between 2^{-4} and 2^{-8} times $\sum \hat{d}_i/n$. When τ is properly selected, the misclassification rate of SC_τ can be reduced from the 18.82% in Table 2 to 4.91%. However, its performance highly depends on this choice. In contrast, for the SLIM, the arbitrary selection of $\tau = 0$ is good enough.

5. Conclusion and Discussioin

We have proposed a new scalable method, SLIM, for detecting communities in networks. The underlying idea is to enhance the information represented by an adjacency matrix by considering a random walk on the network. This method is designed specifically for sparse networks, although it works well for dense networks as well.

The method is stable in the choice of the regularization parameter. This parameter is required to prove consistency in the $E(\text{degree}) = \omega(1)$ sparse scenario. However, simulations suggest that it is not necessary in practice. Therefore, it would be promising to explore whether we can remove the regularization in a theoretical study. Our simulations indicate that the stability of the SLIM with regard to τ may be due to the third step of the SLIM method, namely, the step forcing diagonal entries to zero. Furthermore, simulations and a real-data analysis suggest that methods designed for sparse networks mostly specialize in DCSBM as well. The intrinsic reason for this finding is another interesting topic for future research.

Supplementary Material

The online Supplementary Material contains the proof of Theorem 1 (part A) and additional simulation results (part B) for increasing n.

Acknowledgments

This work was supported by General Research Fund GRF-16305616 and GRF-16304419.

References

Adamic, L. A. and Glance, N. (2005). The political blogosphere and the 2004 US election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery* (Edited by J. Adibi, M. Grobelnik, D. Mladenic, and P. Pantel), 36–43. ACM, New York.

Amini, A. A., Chen, A., Bickel, P. J. and Levina, E. (2013). Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics* **41**, 2097–2122.

Bhattacharyya, S. and Bickel, P. J. (2014). Community detection in networks using graph distance. *arXiv preprint arXiv:1401.3915.*

Chin, P., Rao, A. and Vu, V. (2015). Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery. In *Proceedings of The 28th Conference on Learning Theory* **PMLR 40**, 391–423.

Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis* **21**, 5–30.

Gao, C., Ma, Z., Zhang, A. Y. and Zhou, H. H. (2017). Achieving optimal misclassification proportion in stochastic block models. *The Journal of Machine Learning Research* **18**, 1980–2024.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* **99**, 7821–7826.

Hajek, B., Wu, Y. and Xu, J. (2016). Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory* **62**, 2788–2797.

Jin, J. (2015). Fast community detection by score. *The Annals of Statistics* **43**, 57–89.

Joseph, A. and Yu, B. (2016). Impact of regularization on spectral clustering. *The Annals of Statistics* **44**, 1765–1791.

Kaufman, L. and Rousseeuw, P. J. (1990). Partitioning around medoids (program pam). *Finding Groups in Data: An Introduction to Cluster Analysis* (Edited by L. Kaufman and P. J. Rousseeuw), 68–125. John Wiley & Sons. Inc., Hoboken.

Krzakala, F., Moore, C., Mossel, E., Neeman, J., Sly, A., Zdeborová, L. et al. (2013). Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences* **110**, 20935–20940.

Lei, J. and Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics* **43**, 215–237.

Liu, X., Cheng, H.-M. and Zhang, Z.-Y. (2019). Evaluation of community detection methods. *IEEE Transactions on Knowledge and Data Engineering* **32**, 1736–1746.

Massoulié, L. (2014). Community detection thresholds and the weak ramanujan property. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing* (Edited by D. Shmoys), 694–703. ACM, New York.

Mossel, E., Neeman, J. and Sly, A. (2012). Stochastic block models and reconstruction. *arXiv preprint arXiv:1202.1499.*

Mossel, E., Neeman, J. and Sly, A. (2018). A proof of the block model threshold conjecture. *Combinatorica* **38**, 665–708.

Rohe, K., Chatterjee, S. and Yu, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics* **39**, 1878–1915.

Zhang, A. Y. and Zhou, H. H. (2016). Minimax rates of community detection in stochastic block models. *The Annals of Statistics* **44**, 2252–2280.

Bing-Yi Jing

Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.

E-mail: majing@ust.hk

Ting Li

Department of Applied Mathematics, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

E-mail: tingeric.li@polyu.edu.hk

Ningchen Ying

Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.

E-mail: nying@connect.ust.hk

Xianshi Yu

Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.

E-mail: xyuai@connect.ust.hk

(Received November 2019; accepted May 2020)