Statistics and Its Interface Volume 17 (2024) 159–172

A random projection method for large-scale community detection*

Haobo Qi, Xuening Zhu†, and Hansheng Wang

In this work, we consider a random projection method for a large-scale community detection task. We introduce a random Gaussian matrix that generates several projections on the column space of the network adjacency matrix. The k-means algorithm is then applied with the low-dimensional projected matrix. The computational complexity is much lower than that of the classic spectral clustering methods. Furthermore, the algorithm is easy to implement and accessible for privacy preservation. We can theoretically establish a strong consistency result of the algorithm under the stochastic block model. Extensive numerical studies are conducted to verify the theoretical findings and illustrate the usefulness of the proposed method.

Keywords and phrases: Community detection, Large scale network, Random projection, Stochastic block model.

1. INTRODUCTION

With the rapid development of online network platforms, network data modelling has received great attention. One of the most important tasks is community detection. Community refers to a latent group of network nodes, which are more likely to be connected with each other. In contrast, the nodes from different communities are less likely to form links. For example, people sharing a common preference or coming from the same neighborhood are more likely to interact in social networks. Discovering the community structure has been proven useful in a wide range of scientific fields, including social network analysis [32, 22], financial risk analysis [14], biological studies [25, 24] and many others. This makes community detection one of the most fundamental and interesting topics in network data analysis.

A large amount of literature on community detection has been developed in recent decades. Early approaches were mainly algorithm-based without model assumptions. To name a few, [26] proposed the detection of hierarchical community by using certain types of edge betweeness of network nodes. Later, [27] revised the algorithm by designing an optimal stopping rule based on a novel modularity measure. [7] further improved the computational efficiency of the algorithm to apply it to large-scale networks. The modularity maximization framework inspired many follow-up studies [30, 28, 8]. Another major class of methods has been established under statistical model assumptions. One of the most popular models is the stochastic block model (SBM) proposed by [11]. This model assumes that the nodes from the same community should have a greater probability of being connected. In contrast, the nodes from different communities should have a much smaller likelihood. [17] further characterize the nodes' heterogeneity by devising a degree-corrected stochastic block model (DC-SBM). Another popular statistical model is the latent space model proposed by [10]. This model assumes that the nodes are positioned in a latent space and that they are more likely to connect if their latent positions are closer. To estimate the model parameters (i.e., the communities), both likelihood-based methods [5, 35] and spectral clustering methods [31, 29] are studied. Community detection consistency is established under various model specifications and estimation methods [20, 16, 33].

Despite their usefulness, traditional community detection methods can present great computational challenges, especially for large-scale networks. Consider for example, the largest social network platform in the world: Facebook,. It has more than 1 billion registered users worldwide. Another famous online social platform in mainland China, Sina Weibo, also has more than 100 million active users. For these large-scale networks, the traditional methods suffer greatly from computational costs. Take the modularity method of [27] as an example. As analyzed by [7], its computational complexity is $O(MN + N^2)$, where M and N represent the total number of edges and nodes in the network, respectively. The computational complexity of the spectral clustering method is even higher. This is because the eigenvalue decomposition in spectral clustering consumes a computational complexity of $O(N^3)$. The pseudo likelihood method of [3] typically uses spectral clustering to set its initial value, which makes its computational burden even heavier. The latent space model [10] usually uses Gibbs sampling for posterior estimation, which is also time consuming. As a consequence, how to conduct community detection for large-scale networks in a computationally efficient way becomes a problem of great interest.

*Haobo Qi and Hansheng Wang are supported by National Natural Science Foundation of China (NSFC, 11831008). Xuening Zhu's research is supported by the National Natural Science Foundation of China (nos. 72222009, 11901105, 71991472, U1811461).
†Xuening Zhu is the corresponding author (xueningzhu@fudan.edu.cn).

To address this issue, we propose a random projection method for fast community detection with large-scale networks. The basic idea is to generate a number of random directions in an N-dimensional Euclidean space. Each row of the adjacency matrix is then projected onto these directions. Therefore, their positions in a low-dimensional projected space can be computed. Subsequently, the internode distance can be evaluated for an arbitrary node pair in the projected space. Under appropriate model assumptions (e.g., the SBM model), we prove that the internode distance (after projection) of nodes from different communities should be uniformly larger than that of nodes from the same community with probability tending to 1. This suggests that simple algorithms such as k-means can be employed to consistently discover the community structure. The algorithm is easy to implement and accessible for privacy preservation. Furthermore, a novel eigenvalue ratio criterion is adopted to automatically determine the number of communities. Extensive simulation studies are conducted to demonstrate the method's empirical performances.

The rest of this paper is organized as follows. Section 2 introduces our random projection method for community detection and then discusses its theoretical properties under stochastic block model settings. Section 3 presents several numerical simulation studies to illustrate the finite sample performance of the proposed method and shows its performance in a real-world large-scale network dataset. Section 4 concludes the paper and discusses some interesting future works. All technical details can be found in the Appendix.

2. METHODOLOGY

2.1 The random projection method

Assume there are a total of N nodes in a network, which are indexed by $1 \leq i \leq N$. Their network relationships are described by a network adjacency matrix $A = (a_{ij}) \in \{0,1\}^{N\times N}$, where $a_{ij} = 1$ if the i-th node and j-th node are connected to each other, and $a_{ij} = 0$ otherwise. Let $d_i = \sum_{j=1}^{N} a_{ij}$ denote the degree of nodes i and $D = \mathrm{diag}(d_1, ..., d_N)$. Next, we assume that the network nodes can be grouped into a total of K communities, which are indexed by $1 \leq k \leq K$. Let $C_i \in \{1, 2, ...K\}$ be the community membership of the i-th node. Assume that the community size for the k-th community is N_k as $N_k = \sum_{i=1}^{N} I\{C_i = k\}$ for $1 \leq k \leq K$. We know immediately that $\sum_k N_k = N$. The objective here is to detect the underlying community structure by exploiting the observed network adjacency matrix A.

As previously discussed, various community detection methods have been developed in the literature. However, to the best of our knowledge, it seems that none of those methods can be used to deal with truly large social networks. To solve this problem, we propose a novel random projection method. We first consider an illuminating example. Specifically, suppose that there is a network with only $K = 2$ communities with equal sizes ($N = 2n$ for some $n > 0$). Without loss of generality, we assume that the first n nodes belong to the first community, and the rest belong to the second community. Furthermore, assume that the connecting probabilities within the communities are $p = 1$, and connection probabilities across the communities are $q = 0$. We consider this ideal case for to illustrate our proposal, and we later establish our method in more general settings. To implement our method, we need to first generate a random projection direction as $X = (X_1, ..., X_N)^\top \in \mathbb{R}^N$, where each X_i is randomly generated from, for example, a standard normal distribution. Then, the projected vector can be computed as $Z = (Z_1, Z_2, ..., Z_N)^\top = AX \in \mathbb{R}^N$. Simple calculations reveal that $Z_i = p\sum_{i=1}^{n} X_i$ for $1 \leq i \leq n$ and $Z_i = p\sum_{i=n+1}^{N} X_i$ for $i > n$. We find that the nodes from the same community share identical projected locations. In contrast, the nodes from different communities have different projected locations. This immediately suggests that the community structure can be discovered by a careful study of Z. More specifically, simple algorithms such as k-means can be readily used to serve this purpose.

In practice, the problem could be much more complicated and challenging for the following reasons. First, the intra-community link probability can never be as large as $p = 1$. Instead, it is just a reasonably large number (e.g., $p = 0.5$). This obscures the observed community structure in A, which inevitably makes the projected positions in Z less accurate. Similarly, even though the inter-community link probability q should be very small, it can hardly be as small as exactly 0. That also makes the projected position in Z noisy. Furthermore, the random projection itself also introduces additional noise, which could be large enough to cover the signals in Z. As a result, one single random projection direction might be insufficient. In contrast, multiple random projection directions are necessarily needed.

Inspired by the above discussion, we now propose here a novel random projection method. The detailed algorithm in given in Algorithm 1. Specifically, we take the network adjacency matrix A as an input. Next, with a pre-specified projection dimension d, we generate a random projection matrix as $X = (X_1, ..., X_d) \in \mathbb{R}^{N\times d}$. This leads to the projected position for each node in a d-dimensional Euclidean space as $Z = AX \in \mathbb{R}^{N\times d}$. Typically, we expect $d \ll N$. Here $a_n \ll b_n$ implies $a_n/b_n \to 0$ as $n \to \infty$. This enables us to apply simple algorithms (for example, the k-means algorithm) to Z directly so that the latent community structure can be discovered.

It is noteworthy that the k-means algorithm can be sensitive to the choice of initial cluster centers. To tackle this issue, many improvement algorithms have been developed in the literature. For example, the k-means++ algorithm proposed by [4] uses a novel initialization strategy to stabilize the performance of classical k-means algorithm. In addition, the *kmeans* function in R also provides an option *nstart* for specifying the number of initializations. In our numerical

Algorithm 1 Random Projection Clustering Algorithm

Input: Adjacency matrix $A \in \mathbb{R}^{N\times N}$, latent dimension d, target clustering number K

1: Generate random direction matrix $X \in \mathbb{R}^{N\times d}$ from some probability distribution.
2: Calculate random projection matrix $Z = AX$.
3: Apply the k-means algorithm to the rows of Z.

Output: The clustered label for each node as $\{\widehat{C}_1, ..., \widehat{C}_N\}$.

studies, we use the *kmeans* function in R to implement the experiments and set a large *nstart* number (e.g., 10) to obtain more stable results.

Next, we discuss the computational advantages of the proposed method with respect to computational complexity. As shown in Algorithm 1, the random projection method can be divided into two main steps. The first step is the projection step. That is, the adjacency matrix A is projected in several random directions X into $Z \in \mathbb{R}^{N\times d}$. This leads to a total of $O(N^2d)$ computation operations. However, in practice, the adjacency matrix A can be sparse. As a consequence, it can be stored as a sparse matrix [7] and then computed accordingly. In this case, the computational complexity is mainly determined by the number of edges (rather than the network sizes). Assume the total number of edges is given by $M = \sum_{ij} a_{ij}$. Then, the computational complexity of this projection step can be significantly reduced to $O(Md)$. The second step of our algorithm is the clustering step. This is a standard k-means algorithm operated on d-dimensional features, which consumes a computational complexity of $O(NKd)$. As a consequence, the total computational complexity is $O(Md+NKd)$. As a comparison, the computational complexity for spectral clustering is $O(N^2K + NK^2)$, which could be much higher than that of the proposed method.

2.2 Strong consistency for SBM

As we discussed previously, the proposed random projection method is simple and computationally efficient. However, its statistical properties remain unknown. To address this important question, we next study its theoretical properties under an appropriately assumed model structure. Here, we focus on the arguably most popular model for community detection, that is, the stochastic block model (SBM) studied by [11]. Specifically, we assume an SBM with a total of K communities. We use $C_i \in \{1, 2, , ..., K\}$ as the community membership of the i-th node. Write $P(a_{ij} = 1) = p_{ij}$ and $P = (p_{ij}) \in \mathbb{R}^{N\times N}$. The SBM assumes that $p_{ij} = b_{C_i,C_j}$, where $B = (b_{kl}) \in \mathbb{R}^{K\times K}$ is a $K \times K$ symmetric matrix with each entry $b_{kl} \in [0, 1]$ for $1 \le k, l \le K$. It can be verified that $P = GBG^\top$, where $G = (G_1^\top, ..., G_N^\top) \in \mathbb{R}^{N\times K}$ is a membership matrix with $G_{ik} = 1$ if node i belongs to the k-th community and $G_{ik} = 0$ otherwise. To establish the consistency result of the proposed method, we present a number of important technical conditions as follows.

(A1) (Community Size) Let $n = N/K$; then, there exist two positive constants $0 < c^{(1)}_{\min} < c^{(1)}_{\max}$ such that the community sizes satisfy $c^{(1)}_{\min} n \le \min_k N_k \le \max_k N_k \le c^{(1)}_{\max} n$.

(A2) (Connecting Probability) There exist two positive constants $0 < c^{(2)}_{\min} < c^{(2)}_{\max}$, such that $c^{(2)}_{\min}\theta_n \le \min_k b_{kk} \le \max_k b_{kk} \le c^{(2)}_{\max}\theta_n$, where $n\theta_n \gg \log N$ as $n \to \infty$. Furthermore, we assume that $\sum_{l\neq k} b_{kl} \ll b_{kk}$ as $n \to \infty$ for $k = 1, ..., K$.

The first condition requires different community sizes to be of the same order. This is a standard assumption that has typically been assumed in literature; see, for example, [9] and [12]. This condition enables the subsequent theoretical development to be relatively easier. The second condition defines the key characteristic of a community. That is, the nodes belonging to the same community should be connected with higher probability than nodes from different communities.

We then study the separability of the projected vectors. Ideally, we wish different nodes to be well separated from each other by their projected vectors according to their community membership. Specifically, recall that $Z_i \in \mathbb{R}^d$ is the projected vector from the ith node. Then, nodes from different communities naturally form different groups in terms of Z_i. For each community, we can then compute its community center as $\widehat{\alpha}_k = N_k^{-1}\sum_{C_j=k} Z_j$ for $1 \le k \le K$. Next, we can compute the distance between Z_i and every possible group center as $\delta_{i,k} = \|Z_i - \widehat{\alpha}_k\|^2$. Obviously, we wish the node of interest to stay closer to the community center to which it belongs, compared with other community centers, that is, $\delta_{i,C_i} < \min \delta_{i,k}$ for $k \neq C_i$ and $1 \le i \le N$. That enables us to consider an event set $\mathcal{E} = \{\max_{i,k:C_i=k} \delta_{i,k} < \min_{i,k:C_i\neq k} \delta_{i,k}\}$. It is then of great interest to ask how likely this desirable situation is.

As one can expect, this is a very challenging task. We attempt to address this problem in several steps. In the first step, we study the mean and variance of $\delta_{i,k}$ according to different scenarios. Next, we combine all those preliminary but important findings to form powerful non-asymptotic results to quantify the likelihood of the event of interest $\mathcal{E}$. We start by analyzing $E(\delta_{i,k})$ first. For the simplicity of notations, we assume that $C_i = k$ while the target community changes to k' in the following two propositions.

Proposition 2.1. *Suppose assumptions (A1) and (A2) hold. For a pre-specified dimension d, suppose that $X = (X_{ij}) \in \mathbb{R}^{N\times d}$ with each entry following a standard normal distribution $N(0, 1)$ independently for $1 \le i \le N, 1 \le j \le d$. For arbitrary node i with $C_i = k$, we have*

$$(1)\ E(\delta_{i,k'}) = \begin{cases} dN_k b_{kk} - dN_k b^2_{kk} + o(dn\theta_n^2) \\ \qquad \text{for } k' = k, \\ dN_k b_{kk} + dN_{k'} b^2_{k',k'} + o(dn\theta_n^2) \\ \qquad \text{for } k' \neq k. \end{cases}$$

By Proposition 2.1, we find that for arbitrary node i, the expected distance between the projected vector and its own community center is naturally smaller than that from other communities. Specifically, when $k \neq k'$, the expectation of $\delta_{i,k'}$ shares a common term (i.e. $dN_k b_{kk}$) compared with the case when $k = k'$. More importantly, these two expectations have a difference of term $dN_k b_{kk}^2 + dN_{k'} b_{k',k'}^2$, which is not only determined by community k (i.e., C_i) but also by community k'. This explains why clustering performance could be poor when community sizes are unbalanced or the network has sparse connections. Furthermore, we find that the dimension d plays an important role in (1). It enlarges the gap between $\delta_{i,k'}$s, which gives us better chances to cluster nodes correctly. However, our numerical experiments suggest that a larger projection dimension d also leads to greater variability for the projected positions. Then, whether the gap increase in the mean due to the projection dimension can be offset by the increased variability becomes a critical issue. This inspires the following proposition.

Proposition 2.2. *Suppose the assumptions in Proposition 2.1 holds. We then have*

$$\mathrm{var}(\delta_{i,k'}) = \begin{cases} dN_k^2 b_{kk}^2 + d^2 N_k b_{kk} + o\left(d^2 n\theta_n + dn^2\theta_n^2\right) \\ \qquad\qquad \text{for } k' = k, \\ dN_k^2 b_{kk}^2 + d^2 N_k b_{kk} + o\left(d^2 n\theta_n + dn^2\theta_n^2\right) \\ \qquad\qquad \text{for } k' \neq k. \end{cases}$$

By Proposition 2.2, we find that the variability of the projected vectors increases towards infinity as $n \to \infty$ or $d \to \infty$. A closer look reveals that its leading term is of order $O(d^2 n\theta_n + dn^2\theta_n^2)$ under previous assumptions. This means that the standard deviation of the projected vector is approximately $O(n\theta_n\sqrt{d} + d\sqrt{n\theta_n})$. Recall that the expected distance gap is of order $O(dn\theta_n^2)$, which can dominate the standard deviation when d and n are large. As a consequence, as long as computational resources support it, the projection dimension d should be as large as possible. This is an interesting phenomenon to be numerically demonstrated in the next section. With the help of the previous two propositions, we are then able to establish the following strong consistency results for the community separability result.

Theorem 2.1. (Strong Separability) *Suppose assumptions in Proposition 2.1 hold. Then, we have* $P(\mathcal{E}) \geq 1 - 2\exp\left(-C\min\left\{n\theta_n, d\theta_n^2\right\} + \log K + \log N\right)$, *where* C *is some positive constant.*

In order to achieve the strong consistency of the proposed algorithm, we need $P(\mathcal{E}) \to 1$. Through Theorem 2.1, we can easily find that the driving factor for $P(\mathcal{E}) \to 1$ is the term $\min\{n\theta_n, d\theta_n^2\}$. Then it is of great interest to investigate $n\theta_n$ and $d\theta_n^2$ respectively. First, consider that d is sufficiently large so that $\min\{n\theta_n, d\theta_n^2\} = n\theta_n$. One can find that the separability of the projected vectors is limited by the network itself and cannot be further improved by increasing the projection dimension d. Specifically, we require $n\theta_n \gg \log N$ to ensure $P(\mathcal{E}) \to 1$ as $N \to \infty$. This leads to the signal strength $\theta_n \gg \log N/n$ (as assumed in assumption (A2)), otherwise the proposed method can never achieve strong consistency. Next, we turn to discuss the case that $\min\{n\theta_n, d\theta_n^2\} = d\theta_n^2$. If the signal strength $\theta_n = O(1)$, that is, it does not diminish as $N \to \infty$. Then we only need the projection dimension $d \gg \log N$, which can be a mild condition in practice. When the signal strength θ_n diminishes as $N \to \infty$, to ensure strong consistency we need $d \gg \log N/\theta_n^2$. We consider two special cases. First, suppose that the signal strength is as weak as $\theta_n = \log N/n$. Then we can verify that the projection dimension d should satisfies $d \gg n^2/\log N$. This can be computationally expensive when the network size N is large. Second, suppose that the maximum tolerated projection dimension d satisfies $d = c_0 N$ for some positive constant c_0. Then we can verify that the smallest signal strength to ensure strong consistency should be $\theta_n \gg \sqrt{\log N/N}$. This is a stronger assumption compared with the well known strong consistency results for SBM, that is, $\theta_n \geq c\log N/N$ for some constant $c > 0$ [35, 33]. These two cases imply that when the network becomes sparse, the projected dimension d should be large. This is the price paid for introducing additional noise. In next section we verify our theoretical findings by conducting a number of numerical studies.

3. NUMERICAL STUDIES

3.1 Simulation studies

In this subsection, we conduct a number of simulation studies to evaluate the finite sample performance of our proposed random projection method. We set the connectivity probability matrix as $B = q\mathbf{1}_K\mathbf{1}_K^\top + (p-q)I_K$, where p and q denote the connecting probability between two nodes belonging to the same community and different communities, respectively. This is the standard setting of an SBM with four parameters studied by [31]. Various specification combinations of (N, K, n, d) are studied. For each specification, we randomly repeat the experiment for $T = 100$ times. To evaluate the performance of our proposed method, we consider two clustering performance metrics. The first one is the mis-clustered rate (MCR), which is defined as

$$\mathrm{MCR}(\widehat{\mathcal{C}}, \mathcal{C}) = \frac{\sum_{i=1}^N I(\widehat{C}_i \neq C_i)}{N}.$$

Here $\mathcal{C} = \{C_1, .., C_N\}$ are the ground truth labels and $\widehat{\mathcal{C}} = \{\widehat{C}_1, ..., \widehat{C}_N\}$ are the predicted labels under permutations. As one can see, the smaller the MCR value, the better the clustering result is. The MCR metric is widely used in many community detection studies under SBM, see for example [31, 1, 9]. The second one is the adjusted rand index (ARI), which is a well known metric for assessing the clustering performance [18]. The ARI metric takes value in $[-1, 1]$ and

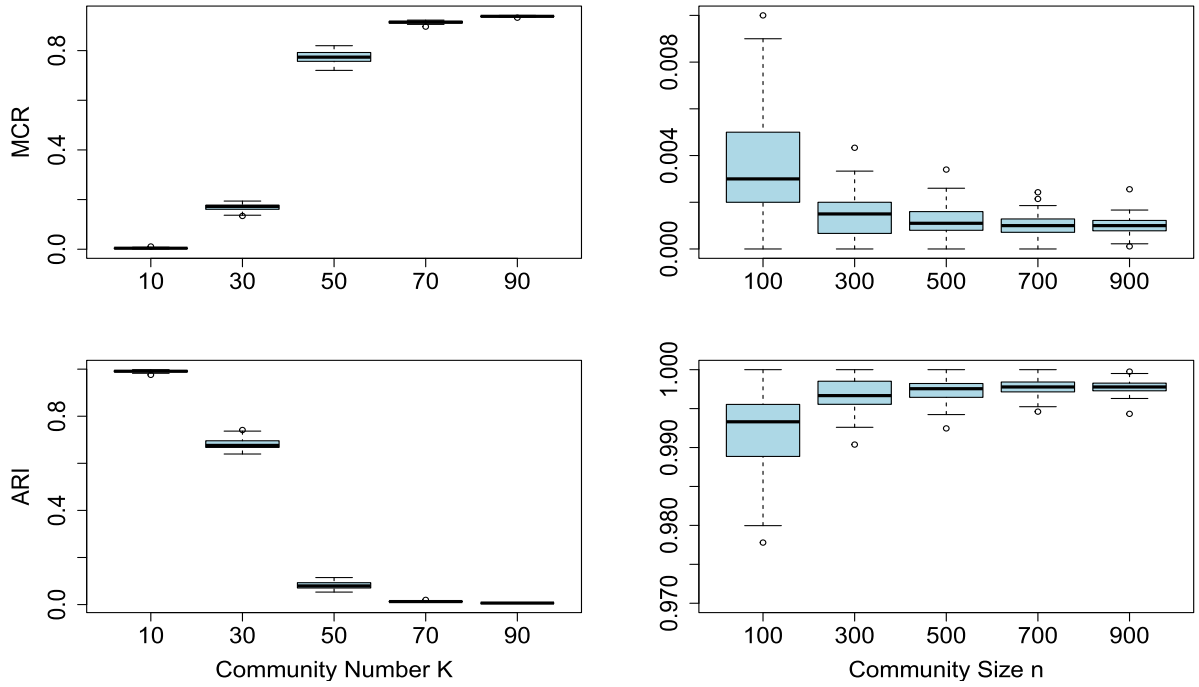


Figure 1. The left two panels, from the top to the bottom, show the MCR and ARI change as the community number K varies from 10 to 90 with a fixed community size $n = 100$, respectively. The right two panels, from the top to the bottom, show the MCR and ARI change as the community size n varies from 100 to 900 with a fixed community number $K = 10$, respectively.

the closer that ARI value to 1, the better the clustering result is. We remark that the ARI can be well defined when the K is not equal to the true cluster number.

We investigate the parameter effects (i.e., K, n and d) of the proposed method in the first two studies. Subsequently, we compare the proposed algorithm with three existing community detection methods and show its computational advance in the third study. Finally, we provide an eigenvalue ratio criterion to select K in the fourth study.

3.2 Effect of community number K and community size n

In our first study, we set $p = 0.5$, $q = 0.05$ and fix $d = 100$. We consider two situations: (i) fix community size $n = 100$ and let K vary from 10 to 90; (ii) fix community number $K = 10$ and let community size n vary from 100 to 900. The detailed results are given in Figure 1. By the left two panels of Figure 1, we find that the clustering performances become worse and worse as the community number K grows for fixed d and n. By the right two panels, we find that the clustering performance become better and better as the community size n grows when we fix the community size K. These results match the claims in Theorem 2.1.

3.3 Effect of projection dimension d

Our second study focuses on the choice of random project dimension d. To this end, we consider the cases that networks are dense and sparse, respectively. In the first case, we consider a dense network with connecting probability $p = 0.2$, $q = 0.01$. We choose the random projection dimension $d_N = (\log N)^2$, which satisfies $d_N \gg \log N$. In the second case, we consider a sparse network with connecting probability $p_N = \log N\sqrt{\log N/N}$ and $q_N = \log N/N$, which satisfies $\theta_n \gg \sqrt{\log N/N}$. As a consequence, the connecting probability diminishes as N increases for the sparse network case. Then we choose the random projection dimension $d_N = 0.1N$. For both two cases, we fix the community number $K = 5$ and let the network size N vary from 1000 to 10000. The detailed results are shown in Figure 2. Similar pattern can be observed in Figure 2, the clustering performances in both cases imply strong consistency as $N \to \infty$. Furthermore, the choice of d_N in case one and the choice of θ_n in case two match the claims and discussions in Theorem 2.1.

3.4 Comparison with existing methods

Our third study concerns about the comparison experiments with several existing methods. Specifically, we consider the Newman-Girvan modularity method [26], the method of latent position cluster model [10] and the spectral cluster method [23]. The Newman-Girvan modularity method (NG) is implemented by the function *cluster_edge_betweenness* in R package *igraph*[1]. The latent position cluster model (LPCM) is implemented via R pack-

[1] https://cran.r-project.org/web/packages/igraph/index.html.

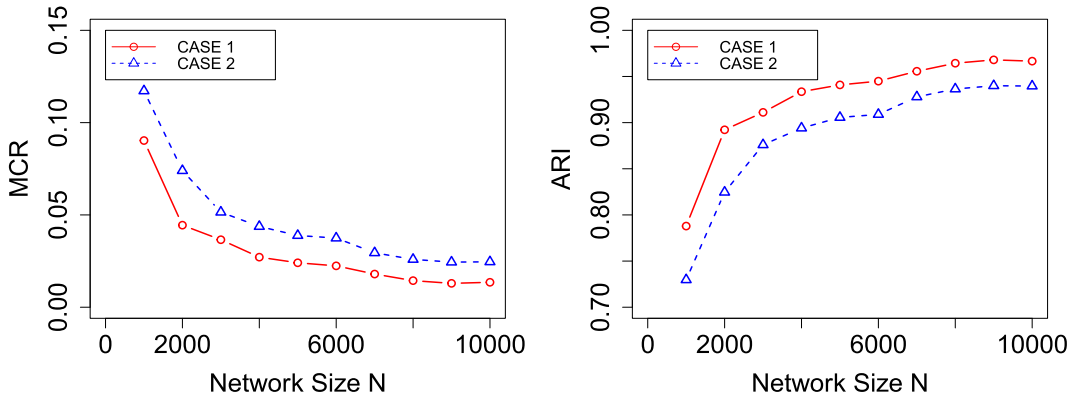


Figure 2. The left panel shows the MCR changes of two cases as network size N varies from 1000 to 10000 with fixed community number $K = 5$; The right panel shows the ARI changes of two cases as network size N varies from 1000 to 10000 with fixed community number $K = 5$.

age *latentnet*[2], which is provided by [10]. We consider the Euclidean distance in a five dimensional latent space and set the *burnin* = 20,000. Moreover, the spectral clustering (SC) method is implemented by the R package *RSpectra*[3], which is designed for large-scale eigenvalue decomposition. As for the network settings, we set $p = 0.5$, $q = 0.2$, fix the number of community $K = 3$, and let the community size n vary from 10 to 1000. For the proposed random projection method, we fix $d = 200$ for experiments with $K = 2$ and $d = 300$ for experiments with $K = 5$. The clustering performances are evaluated by the mean mis-clustered rate (MCR), adjusted rand index (ARI) and CPU time (in seconds). Note that the NG method does not take a prespecified clustering number, therefore the cluster number K might be over-estimated (or under-estimated). Thus we omit the MCR value for the NG method since it is not well defined in this case. We further omit the results that need CPU time more that 600 seconds.

According to the results in Table 1, we can draw the following conclusions. First, almost all the methods have better clustering accuracy when community size n diverges with a fixed community number K. The only exceptional case occurs for the Newman-Girvan modularity method when $K = 5$. This is because the NG method tends to overestimate the cluster number and thus introducing extra clustering instability. Second, compared to other methods, the proposed RP method is obviously more computationally efficient with comparable clustering accuracy. This advantage increases as the network size becomes larger. For instance, when $K = 5$ and $n = 500$, the RP method consumes 0.2 seconds while SC method consumes 23.5 seconds, which is 100 times larger than the RP methods. Moreover, both NG and LPCM methods are not able to produce the result within 600 seconds under this circumstance.

[2]https://sites.stat.washington.edu/raftery/Research/latentnet.html.
[3]https://cran.r-project.org/web/packages/RSpectra/vignettes/introduction.html.

3.5 Selection of community number K

For the last simulation study, we try to evaluate the empirical performance of an intuitive and simple method for estimating the community number K. There exists rich literature for estimating community number K; see, for example, [15], [6], [19] [21], [12]. In addition to these works, we apply a simple method of the maximum eigenvalue ratio criterion. Recall that probability matrix P is of rank K under the SBM setting; we should expect matrix $E = X^\top A^\top AX/(Nd) \in \mathbb{R}^d (K < d \ll N)$ to have a K large top eigenvalue, while the rest are comparatively small. Specifically, let $\widehat{\lambda}_1 \geq \widehat{\lambda}_1 \cdots \geq \widehat{\lambda}_d \geq 0$ be the eigenvalues of matrix E. Thus, if we define an eigenvalue ratio criterion as $w_k = \widehat{\lambda}_k / \widehat{\lambda}_{k+1}$ for $1 \leq k \leq d-1$, we should expect w_k to reach its maximum at $k = K$. As a result, we choose the estimated community number $\widehat{K}$ as $\widehat{K} = \text{argmax}_k \, w_k$. We set $p = 0.2, q = 0.01$ and $n = N/K = 1000$. To evaluate the numerical performance of this maximum eigenvalue ratio criterion, we set $K = 5, 10, 20$ and let d vary from 10 to 100 to investigate the effective dimension d. The results are summarized in Table 2, and Figure 3 gives a visualized illustration of the maximum eigenvalue ratio criterion.

3.6 Real data examples

In this subsection, we consider two real-world network datasets to illustrate the accuracy and computational efficiency of our method. The first is the political blog network collected by [2]. The network consists of 1,490 blogs about US politics where the edges refer to web links. All blogs are labeled with 0 for liberal and 1 for conservative. This leads to two ground truth communities. However, straightforward community detection leads to poor clustering performance due to the imbalance and sparsity of the network. To this end, we only use the largest connected subnetwork, which contains 1,222 nodes with community sizes 586 and 636 for liberal and conservative, respectively. We applied our random projection method and the classic spectral clustering

Table 1. Comparisons of four community detection methods. The reported CPU time results are multiplied by 10 and those more than 600 seconds are represented by '−'.

		RP			NG			LPCM			SC		
K	n	MCR	ARI	Time	MCR	ARI	Time	MCR	ARI	Time	MCR	ARI	Time
	10	0.303	0.188	0.005	–	0.354	0.020	0.290	0.353	119.6	0.390	0.091	0.007
	20	0.075	0.716	0.007	–	0.566	0.324	0.020	0.920	339.1	0.040	0.845	0.010
	50	0.017	0.934	0.023	–	0.941	25.64	0.004	0.984	2303	0.006	0.975	0.021
2	100	0.004	0.984	0.044	–	0.998	872.7	–	–	–	0	1	0.110
	200	0.002	0.994	0.132	–	–	–	–	–	–	0	1	0.635
	500	0.001	0.997	0.908	–	–	–	–	–	–	0	1	9.528
	10	0.584	0.081	0.009	–	0.129	0.706	0.540	0.173	2022.0	0.600	0.083	0.011
	20	0.469	0.231	0.022	–	0.079	15.03	–	–	–	0.390	0.845	0.033
	50	0.047	0.887	0.069	–	0.065	1356	–	–	–	0.011	0.975	0.210
5	100	0.011	0.972	0.192	–	–	–	–	–	–	0.005	0.996	1.288
	200	0.002	0.994	0.387	–	–	–	–	–	–	0.001	0.999	10.96
	500	0.001	0.998	2.210	–	–	–	–	–	–	0	1	235.1

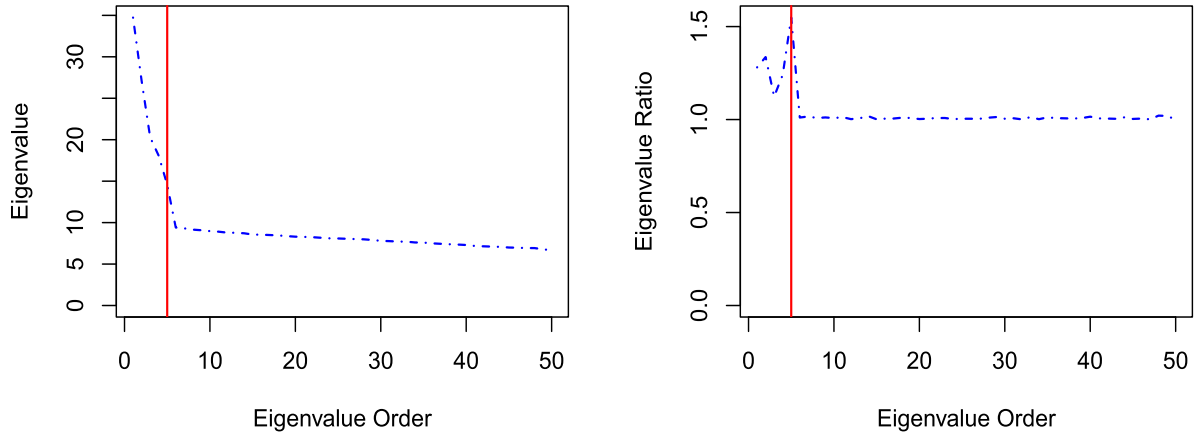


Figure 3. The left panel shows the eigenvalue of matrix $E = X^\top A^\top AX/(Nd)$ (of one replicate) in descending order, and the red vertical line shows the true value of K. The right panel shows the eigenvalue ratio of matrix E, and the red vertical line shows the true value of K.

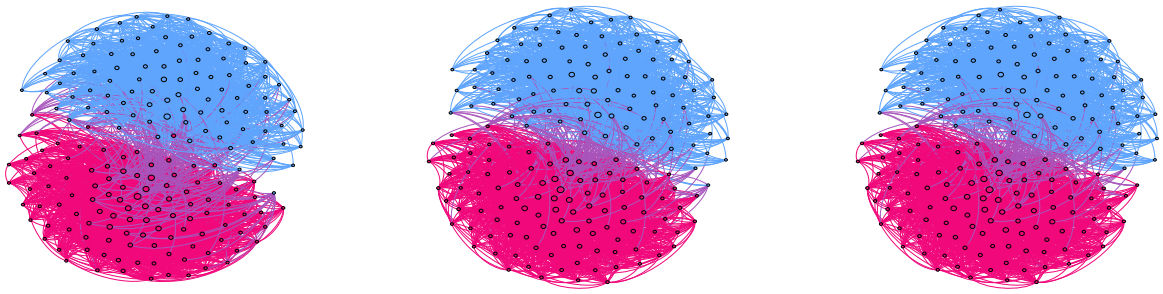
Table 2. Accuracy of the estimation of K based on the maximum eigenvalue ratio criterion under different (K, d) specification combinations when fixing $n = 1000$.

K \ d	5	10	20	50	75	100
3	10%	54%	96%	100%	100%	100%
5	-	2%	88%	100%	100%	100%
10	-	-	2%	100%	100%	100%
20	-	-	-	18%	94%	96%

method to these network data. Figure 4 shows the clustering results of 200 nodes with in-degrees no less than 30. We find that the performance of the random projection method is comparable with that of the classic spectral clustering method but with lower computation time. Specifically, the spectral clustering method requires 0.5616 s to complete the task, while the random projection method (with $d = 50$) only takes 0.2695 s to achieve the same clustering accuracy.

The second real-world network dataset is the Sina Weibo network data. The dataset is collected from Sina Weibo (www.weibo.com), which is arguably the largest Twitter-type social media platform in China [13]. After the basic data cleaning procedure, we keep $N = 1{,}153$ nodes with in-degrees larger than 30. The number of edges is $M = 68{,}109$. We then apply our random projection method to this network with $d = 100$. The eigenvalue ratio criterion suggests $K = 6$ as the number of communities. The clustering results can be found in the left panel of Figure 5. The within-community density is $n_1^{-1}\sum_{i,j} a_{ij} I(\widehat{C}_i = \widehat{C}_j) = 0.1529$, and the between-community density is $n_2^{-1}\sum_{i,j} a_{ij} I(\widehat{C}_i \neq \widehat{C}_j) = 0.01891$, where $n_1 = \sum_k N_k(N_k - 1)$ and $n_2 = N(N-1) - \sum_k N_k(N_k - 1)$.

To explore the information behind the community detection result, we further illustrate the clustering result with several nodal covariates. Specifically, the dataset contains



(a) Ground Truth (b) Random Projection (c) Spectral Clustering

Figure 4. Clustering results of the top 200 nodes with in-degrees larger than 30 from the political blogs dataset via (b) random projection ($d = 100$) and (c) spectral clustering compared with (a) ground truth.

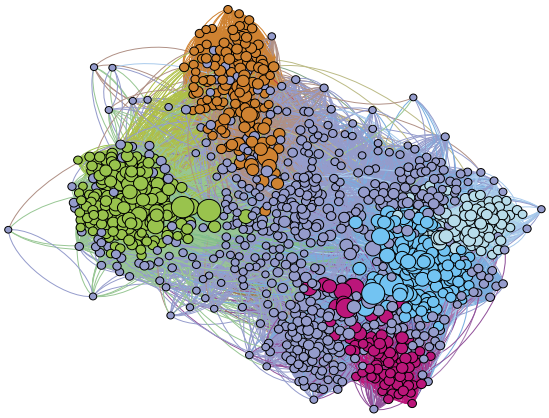


Figure 5. Clustering results of the Sina Weibo dataset via the random projection method with $K = 6$. Different colors represent different communities, and point sizes represent the degree of nodes.

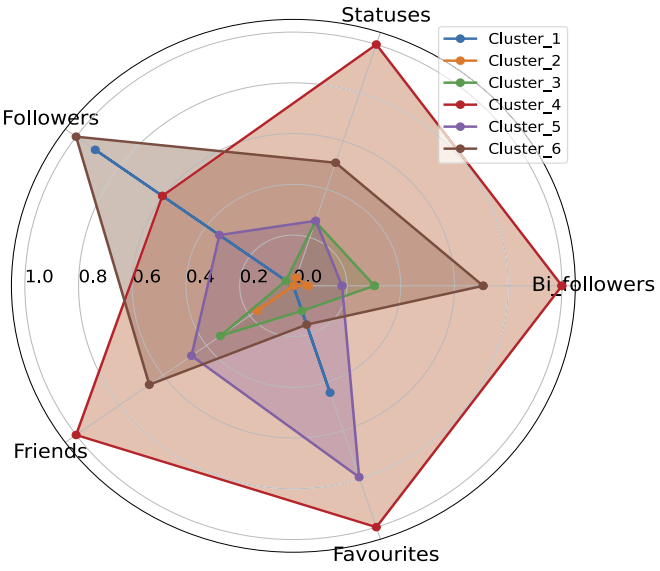


Figure 6. Radar plot of the mean value of each attribute within all $K = 6$ communities. The scale of each attribute is normalized to $[0, 1]$ by the maximum and minimum.

five covariates of the Weibo users. These covariates are bi-follower count, status count, follower count, friend count and favorite count. We calculate the mean value of each covariate within all six communities, as shown in the radar plot in Figure 6. The distributions of covariates are quite different across different communities. This reveals that our community detection result is informative and helpful for further explorations. Furthermore, this implies that the covariate information may further help community detection, and we discuss it as a future study topic in the next section.

4. CONCLUSIONS

In this paper, we propose a simple random projection method for large-scale network community detection. The basic idea is to generate a number of random directions in an N-dimensional Euclidean space. Each row of the adjacency matrix is then projected onto these directions. Therefore, their positions in a low-dimensional projected space can be computed. Subsequently, k-means can be applied to these random projections to determine the community structure. Furthermore, we adopt a simple eigenvalue ratio criterion on the random projections so that the community number can be determined automatically. Our proposed method has the following advantages. First, it is simple, feasible and computationally cheap for large-scale network data. Second, our method preserves privacy with the help of random projection. We do not need to actually obtain the adjacency matrix A once the random projections are calculated. Third, our random projection method is naturally suitable for parallel computing, which makes it more flexible to deal with large-scale network data or widely distributed stored network data.

To conclude this work, we will discuss several interesting directions for future study. First, the statistical properties of our proposed method are studied under standard SBM. As an algorithm-based method, it will be of great interest to study its theoretical properties under other network structures, including degree-corrected SBM and latent space model. Second, the random projection we used in this work is a standard normal random matrix. How the covariance matrix influences the result remains unknown. Furthermore, we can use a random projection matrix with other distributions. For instance, when we use the Bernoulli variables to form a projection matrix, the random projection becomes a subsampling result. This may unify all these methods under the random projection framework. Finally, our method is naturally suited for parallel computing, and will be of great interest to develop a corresponding method for parallel or distributed community detection methods.

APPENDIX A. APPENDIX SECTION

A.1 Proof of Proposition 2.1

Define $\overline{A}_{(k)} = N_k^{-1}\sum_{C_j=k} A_j$, where $A_i^\top = (a_{i1}, ..., a_{iN})$ denotes the i-th row of adjacency matrix A. Define $\Omega_{i,k'} = (A_i - \overline{A}_{(k')})(A_i - \overline{A}_{(k')})^\top$ Note that $Z_i = X^\top A_i$, we then have $\delta_{i,k'} = \|Z_i - \widehat{\alpha}_{k'}\|^2 = \mathrm{tr}(X^\top \Omega_{i,k'} X) = \sum_{s=1}^d X_s^\top \Omega_{i,k'} X_s$. Then we have

$$\begin{aligned} E(\delta_{i,k'}) &= E\left(\sum_{s=1}^d X_s^\top \Omega_{i,k'} X_s\right) \\ &= \sum_{s=1}^d E\left\{E\left(X_s^\top \Omega_{i,k'} X_s \middle| A\right)\right\} = dE\{\mathrm{tr}(\Omega_{i,k'})\} \end{aligned}$$

Then it suffices to calculate $E\{\mathrm{tr}(\Omega_{i,k'})\}$, we will consider two cases as $k' = k$ and $k' \neq k$, respectively.

Case 1: $k' = k$

Under this case, we know that $i \in \{j : C_j = k'\}$, thus we have

$$\begin{aligned} \mathrm{tr}(\Omega_{i,k'}) &= \sum_{t=1}^N \left(a_{it} - \frac{1}{N_k}\sum_{j:C_j=k} a_{jt}\right)^2 \\ &= \frac{1}{N_k^2}\sum_{t=1}^N \left\{\sum_{j:C_j=k, j\neq i} (a_{it} - a_{jt})\right\}^2 \\ &= \frac{1}{N_k^2}\sum_{t=1}^N \left\{\sum_{j:C_j=k, j\neq i} (a_{it} - a_{jt})^2 \right. \\ &\qquad \left. + \sum_{\substack{j,r:C_j=k, C_r=k \\ j\neq r, j\neq i, r\neq i}} (a_{it} - a_{it}a_{rt} - a_{it}a_{jt} + a_{rt}a_{jt})\right\} \end{aligned}$$

It can be carefully verified that $E(a_{it} - a_{jt})^2 = 2b_{kl}(1-b_{kl})$ when $C_t = l$. Similarly, we have $E(a_{it} - a_{it}a_{rt} - a_{it}a_{jt} + a_{rt}a_{jt}) = b_{kl}(1-b_{kl})$ when $C_t = l$. As a result, we have

$$\begin{aligned} E[\mathrm{tr}(\Omega_{i,k'})] &= \frac{N_k - 1}{N_k}\sum_{l=1}^K N_l b_{kl}(1-b_{kl}) \\ &= N_k b_{kk}(1-b_{kk}) + O\left(\sum_{l\neq k} N_l b_{kl}\right) \end{aligned} \tag{2}$$

Case 2: $k' \neq k$

Under this case, we know that $i \notin \{j : C_j = k'\}$, thus we have

$$\begin{aligned} \mathrm{tr}(\Omega_{i,k'}) &= \sum_{t=1}^N \left(a_{it} - \frac{1}{N_{k'}}\sum_{j:C_j=k} a_{jt}\right)^2 \\ &= \frac{1}{N_{k'}^2}\sum_{t=1}^N \left\{\sum_{j:C_j=k'} (a_{it} - a_{jt})\right\}^2 \\ &= \frac{1}{N_{k'}^2}\sum_{t=1}^N \left\{\sum_{j:C_j=k'} (a_{it} - a_{jt})^2 \right. \\ &\qquad \left. + \sum_{\substack{j,r:j\neq r \\ C_j=k', C_r=k'}} (a_{it} - a_{it}a_{rt} - a_{it}a_{jt} + a_{rt}a_{jt})\right\} \end{aligned}$$

We can calculate that $E(a_{it} - a_{jt})^2 = E(a_{it} - 2a_{it}a_{jt} + a_{jt}) = b_{kl}(1-b_{k'l}) + b_{k'l}(1-b_{kl})$ when $C_t = l$. Similarly, we have $E(a_{it} - a_{it}a_{rt} - a_{it}a_{jt} + a_{rt}a_{jt}) = b_{kl}(1-b_{k'l})^2 + b_{k'l}^2(1-b_{kl})$ when $C_t = l$. Then we have

$$\begin{aligned} & E[\mathrm{tr}(\Omega_{i,k'})] \\ &= \frac{1}{N_{k'}^2}\sum_{t=1}^N \left\{\sum_{j:C_j=k'} E(a_{it} - a_{jt})^2 \right. \\ &\qquad \left. + \sum_{\substack{j,r:j\neq r \\ C_j=k', C_r=k'}} (a_{it} - a_{it}a_{rt} - a_{it}a_{jt} + a_{rt}a_{jt})\right\} \\ &= \frac{1}{N_{k'}^2}\sum_{l=1}^K \sum_{C_t=l} \left\{\sum_{j:C_j=k'} E(a_{it} - a_{jt})^2 \right. \\ &\qquad \left. + \sum_{\substack{j,r:j\neq r \\ C_j=k', C_r=k'}} (a_{it} - a_{it}a_{rt} - a_{it}a_{jt} + a_{rt}a_{jt})\right\} \\ &= \sum_{l=1}^K \sum_{C_t=l} \left\{\frac{1}{N_{k'}}[b_{kl}(1-b_{k'l}) + b_{k'l}(1-b_{kl})]\right. \end{aligned}$$

$$
\begin{aligned}
&+\frac{N_{k'}-1}{N_{k'}}[b_{kl}(1-b_{k'l})^2+b_{k'l}^2(1-b_{kl})]\Big\}\\
=&\sum_{l=1}^{K}\Big\{\frac{N_l}{N_{k'}}[b_{kl}(1-b_{k'l})+b_{k'l}(1-b_{kl})]\\
&+\frac{N_{k'}-1}{N_{k'}}N_l[b_{kl}(1-b_{k'l})^2+b_{k'l}^2(1-b_{kl})]\Big\}\\
=&\sum_{l=1}^{K}\{N_l[b_{kl}(1-b_{k'l})^2+b_{k'l}^2(1-b_{kl})]\}\\
&+\sum_{l=1}^{K}\left\{\frac{N_l}{N_{k'}}b_{k'l}(1-b_{k'l})\right\}\\
=&\ N_kb_{kk}+N_{k'}b_{k',k'}^2+O\left(\sum_{l\neq k}N_lb_{kl}\right) \qquad (3)
\end{aligned}
$$

Furthermore, we can calculate that the difference between $E(\delta_{i,k'})$s for $i\in\{j:C_j=k'\}$ and $i\notin\{j:C_j=k'\}$ is $N_kb_{kk}^2+N_{k'}b_{k',k'}^2+O\big(\sum_{l\neq k}N_lb_{kl}^2\big)$. This finishes the proof.

A.2 Proof of Proposition 2.2

Recall that $\delta_{i,k'}=\|Z_i-\widehat{\alpha}_{k'}\|^2=\mathrm{tr}(X^\top\Omega_{i,k'}X)=\sum_{s=1}^d X_s^\top\Omega_{i,k'}X_s$. Conditional on A, we can verify that $X_s^\top\Omega_{i,k'}X_s\sim\mathrm{tr}(\Omega_{i,k'})\chi^2(1)$ for $s=1,2,...,d$. Note that $X_1,...,X_d$ are independent, we have $\sum_{s=1}^d X_s^\top\Omega_{i,k'}X_s\sim\mathrm{tr}(\Omega_{i,k'})\chi^2(d)$. Then we have

$$
\begin{aligned}
&\mathrm{var}(\delta_{i,k'})=\mathrm{var}\left(\sum_{s=1}^d X_s^\top\Omega_{i,k'}X_s\right)\\
&=E\left\{\mathrm{var}\left(\sum_{s=1}^d X_s^\top\Omega_{i,k'}X_s\Big|A\right)\right\}\\
&\quad+\mathrm{var}\left\{E\left(\sum_{s=1}^d X_s^\top\Omega_{i,k'}X_s\Big|A\right)\right\}\\
&=2dE\{\mathrm{tr}^2(\Omega_{i,k'})\}+d^2\,\mathrm{var}\{\mathrm{tr}(\Omega_{i,k'})\}
\end{aligned}
$$

Then it suffices to calculate $E\{\mathrm{tr}^2(\Omega_{i,k'})\}$ and $\mathrm{var}\{\mathrm{tr}(\Omega_{i,k'})\}$. We first consider $E\{\mathrm{tr}^2(\Omega_{i,k'})\}$.

$$
\begin{aligned}
&\mathrm{tr}^2(\Omega_{i,k'})=\left\{\sum_{t=1}^N\left(a_{it}-N_{k'}^{-1}\sum_{j:C_j=k'}a_{jt}\right)^2\right\}^2\\
&=\sum_{t=1}^N\left(a_{it}-N_{k'}^{-1}\sum_{j:C_j=k'}a_{jt}\right)^4\\
&\quad+\sum_{s=1}^N\sum_{t\neq s}\left(a_{it}-N_{k'}^{-1}\sum_{j:C_j=k'}a_{jt}\right)^2\left(a_{is}-N_{k'}^{-1}\sum_{j:C_j=k'}a_{js}\right)^2 \qquad (4)
\end{aligned}
$$

We then discuss the above two parts respectively.

Case 1: $k'=k$

Under this case, we know that $i\in\{j:C_j=k'\}$, then we have

$$
\begin{aligned}
&E\left(a_{it}-N_k^{-1}\sum_{j:C_j=k}a_{jt}\right)^4\\
&=\frac{1}{N_k^4}\{(N_k-1)\Delta_{1t}+4(N_k-1)(N_k-2)\Delta_{2t}\\
&+3(N_k-1)(N_k-2)\Delta_{3t}+6(N_k-1)(N_k-2)(N_k-3)\Delta_{4t}\\
&+(N_k-1)(N_k-2)(N_k-3)(N_k-4)\Delta_{5t}\},
\end{aligned}
$$

where $\Delta_{1t}=E(a_{it}-a_{jt})^4,\Delta_{2t}=E\{(a_{it}-a_{jt})^3(a_{it}-a_{rt})\},\Delta_{3t}=E\{(a_{it}-a_{jt})^2(a_{it}-a_{rt})^2\},\Delta_{4t}=E\{(a_{it}-a_{jt})^2(a_{it}-a_{rt})(a_{it}-a_{mt})\},\Delta_{5t}=E\{(a_{it}-a_{jt})(a_{it}-a_{rt})(a_{it}-a_{mt})(a_{it}-a_{ot})\}$. It can be calculated that $\Delta_{1t}=2b_{kl}(1-b_{kl})$, $\Delta_{2t}=b_{kl}(1-b_{kl})$, $\Delta_{3t}=b_{kl}(1-b_{kl})^2+b_{kl}^2(1-b_{kl})$, $\Delta_{4t}=b_{kl}(1-b_{kl})^3+b_{kl}^3(1-b_{kl})$ and $\Delta_{5t}=b_{kl}(1-b_{kl})^4+b_{kl}^4(1-b_{kl})$ when $C_t=l$. Combine the results above, we have

$$
\begin{aligned}
&E\left(a_{it}-N_k^{-1}\sum_{j:C_j=k}a_{jt}\right)^4\\
&=b_{kl}(1-b_{kl})^4+b_{kl}^4(1-b_{kl})+O\left(\frac{b_{kl}}{N_k}\right)\\
&=b_{kl}+O\left(b_{kl}^2+\frac{b_{kl}}{N_k}\right)
\end{aligned}
$$

Then we have

$$
\begin{aligned}
&\sum_{t=1}^N E\left(a_{it}-N_k^{-1}\sum_{j:C_j=k}a_{jt}\right)^4\\
&=N_kb_{kk}+O(N_kb_{kk}^2+b_{kk}+\sum_{l\neq k}N_lb_{kl})
\end{aligned}
$$

On the other hand, the second term in equation (4) is the summation of $(a_{it}-N_k^{-1}\sum_{j:C_j=k}a_{jt})^2$, whose expectation has been studied in (2) for $C_i=k=k'$. Recall that $E(a_{it}-N_k^{-1}\sum_{j:C_j=k}a_{jt})^2=[(N_k-1)/N_k]b_{kl}(1-b_{kl})$ when $C_t=l$. We then can calculate that

$$
\begin{aligned}
&E\left\{\sum_{t=1}^N\sum_{t\neq s}\left(a_{it}-\sum_{j:C_j=k}\frac{a_{jt}}{N_k}\right)^2\left(a_{is}-\sum_{j:C_j=k}\frac{a_{js}}{N_k}\right)^2\right\}\\
&=\left\{\sum_{t=1}^N E\left(a_{it}-N_k^{-1}\sum_{j:C_j=k}a_{jt}\right)^2\right\}^2\\
&\quad-\sum_{t=1}^N\left\{E\left(a_{it}-N_k^{-1}\sum_{j:C_j=k}a_{jt}\right)^2\right\}^2
\end{aligned}
$$

$$= \left(\frac{N_k-1}{N_k}\right)^2 \left\{\sum_{l=1}^{K} N_l b_{kl}(1-b_{kl})\right\}^2 - \left(\frac{N_k-1}{N_k}\right)^2 \sum_{l=1}^{K} N_l b_{kl}^2(1-b_{kl})^2$$

Combine all the results above, we have

$$E\{\mathrm{tr}^2(\Omega_{i,k'})\} = E\left\{\sum_{t=1}^{N}\left(a_{it} - N_k^{-1}\sum_{C_j=k} a_{jt}\right)^2\right\}^2$$
$$= \sum_{t=1}^{N} E\left(a_{it} - N_k^{-1}\sum_{j:C_j=k} a_{jt}\right)^4 + \sum_{t=1}^{N}\sum_{t\neq s} E\left(a_{it} - N_k^{-1}\sum_{j:C_j=k} a_{jt}\right)^2 \times E\left(a_{is} - N_k^{-1}\sum_{j:C_j=k} a_{js}\right)^2$$
$$= \sum_{l=1}^{K} N_l[b_{kl} + O(b_{kl}^2)] + \left(\frac{N_k-1}{N_k}\right)^2\left\{\sum_{l=1}^{K} N_l b_{kl}(1-b_{kl})\right\}^2 - \left(\frac{N_k-1}{N_k}\right)^2 \sum_{l=1}^{K} N_l b_{kl}^2(1-b_{kl})^2$$
$$= N_k^2 b_{kk}^2 + O\left(N_k b_{kk} + \sum_{l\neq k} N_l^2 b_{kl}^2\right)$$

The variance can be calculated subsequently as

$$\mathrm{var}\{\mathrm{tr}(\Omega_{i,k'})\} = E\{\mathrm{tr}^2(\Omega_{i,k'})\} - E\{\mathrm{tr}(\Omega_{i,k'})\}^2$$
$$= \sum_{t=1}^{N} E\left(a_{it} - N_k^{-1}\sum_{j:C_j=k} a_{jt}\right)^4 - \sum_{t=1}^{N}\left\{E\left(a_{it} - N_k^{-1}\sum_{j:C_j=k} a_{jt}\right)^2\right\}^2$$
$$= \sum_{l=1}^{K} N_l[b_{kl} + O(b_{kl}^2)] - \left(\frac{N_k-1}{N_k}\right)^2 \sum_{l=1}^{K} N_l b_{kl}^2(1-b_{kl})^2 \quad (5)$$
$$= N_k b_{kk} + O\left(N_k b_{kk}^2 + \sum_{l\neq k} N_l b_{kl}\right)$$

Case 2: $k' \neq k$

Under this case, we know that $i \notin \{j : C_j = k'\}$, then we have

$$E\left(a_{it} - N_{k'}^{-1}\sum_{j:C_j=k'} a_{jt}\right)^4$$
$$= \frac{1}{N_{k'}^4}\{N_{k'}\Delta_{1t} + 3N_{k'}(N_{k'}-1)\Delta_{2t} + 4N_{k'}(N_{k'}-1)\Delta_{3t} + 6N_{k'}(N_{k'}-1)(N_{k'}-2)\Delta_{4t} + N_{k'}(N_{k'}-1)(N_{k'}-2)(N_{k'}-3)\Delta_{5t}\},$$

where $\Delta_{1t}, \Delta_{2t}, ..., \Delta_{5t}$ are defined same as the case for $k' = k$. It can be calculated that $\Delta_{1t} = b_{kl}(1-b_{k'l}) + b_{k'l}(1-b_{kl})$, $\Delta_{2t} = b_{kl}(1-b_{k'l})^2 + b_{k'l}^2(1-b_{kl})$, $\Delta_{3t} = b_{kl}(1-b_{k'l})^2 + b_{k'l}^2(1-b_{kl})$, $\Delta_{4t} = b_{kl}(1-b_{k'l})^3 + b_{k'l}^3(1-b_{kl})$ and $\Delta_{5t} = b_{kl}(1-b_{k'l})^4 + b_{k'l}^4(1-b_{kl})$. Combine the results above, we have

$$E\left(A_{it} - N_{k'}^{-1}\sum_{C_j=k'} A_{jt}\right)^4$$
$$= \frac{1}{N_{k'}^4}\{N_{k'}\Delta_{1t} + 3N_{k'}(N_{k'}-1)\Delta_{2t} + 4N_{k'}(N_{k'}-1)\Delta_{3t} + 6N_{k'}(N_{k'}-1)(N_{k'}-2)\Delta_{4t} + N_{k'}(N_{k'}-1)(N_{k'}-2)(N_{k'}-3)\Delta_{5t}\}$$
$$= b_{kl}(1-b_{k'l})^4 + b_{k'l}^4(1-b_{kl}) + O\left(\frac{b_{kl}}{N_k}\right)$$
$$= b_{kl} + O\left(b_{k'l}^4 + \frac{b_{kl}}{N_{k'}}\right)$$

Then we have

$$\sum_{t=1}^{N} E\left(a_{it} - N_{k'}^{-1}\sum_{j:C_j=k'} a_{jt}\right)^4$$
$$= N_k b_{kk} + O\left(N_{k'} b_{k',k'}^4 + b_{kk} + \sum_{l\neq k} N_l b_{kl}\right)$$

Similarly, according to (3) Appendix A.1 we have $E(a_{it} - N_{k'}^{-1}\sum_{j:C_j=k'} a_{jt})^2 = \{N_l[b_{kl}(1-b_{k'l})^2 + b_{k'l}^2(1-b_{kl})]\} + O(b_{k'l})$ when $C_t = l$. We then can calculate that

$$E\left\{\sum_{t=1}^{N}\sum_{s\neq t}\left(a_{it} - \sum_{j:C_j=k'}\frac{a_{jt}}{N_{k'}}\right)^2\left(a_{is} - \sum_{j:C_j=k'}\frac{a_{js}}{N_{k'}}\right)^2\right\}$$
$$= \left\{\sum_{t=1}^{N} E\left(a_{it} - N_{k'}^{-1}\sum_{j:C_j=k'} a_{jt}\right)\right\}^2 - \sum_{t=1}^{N}\left\{E\left(a_{it} - N_{k'}^{-1}\sum_{j:C_j=k'} a_{jt}\right)^2\right\}^2$$

$$= \left\{\sum_{l=1}^{K}\left\{N_l[b_{kl}(1-b_{k'l})^2 + b_{k'l}^2(1-b_{kl})] + O(b_{k'l})\right\}\right\}^2$$
$$- \sum_{l=1}^{K} N_l[b_{kl}(1-b_{k'l})^2 + b_{k'l}^2(1-b_{kl})]^2 + O\left(\sum_{l\neq k'} N_l b_{k'l}\right)$$

Combine all the results above, we have

$$E\{\mathrm{tr}^2(\Omega_{i,k'})\} = E\left\{\sum_{t=1}^{N}\left(a_{it} - N_{k'}^{-1}\sum_{j:C_j=k'} a_{jt}\right)^2\right\}^2$$
$$= \sum_{t=1}^{N} E\left(a_{it} - N_{k'}^{-1}\sum_{j:C_j=k'} a_{jt}\right)^4$$
$$+ \sum_{t=1}^{N}\sum_{s\neq t} E\left(a_{it} - \sum_{j:C_j=k'}\frac{a_{jt}}{N_{k'}}\right)^2 E\left(a_{is} - \sum_{j:C_j=k'}\frac{a_{js}}{N_{k'}}\right)^2$$
$$= N_k^2 b_{kk}^2 + O\left(N_k b_{kk} + \sum_{l\neq k}^{K} N_l^2 b_{kl}^2\right)$$

The variance can be calculated subsequently as

$$\mathrm{var}\{\mathrm{tr}(\Omega_{i,k'})\} = E\{\mathrm{tr}^2(\Omega_{i,k'})\} - E\{\mathrm{tr}(\Omega_{i,k'})\}^2$$
$$= \sum_{t=1}^{N} E\left(a_{it} - N_{k'}^{-1}\sum_{j:C_j=k'} a_{jt}\right)^4$$
$$- \sum_{t=1}^{N}\left\{E\left(a_{it} - N_{k'}^{-1}\sum_{j:C_j=k'} a_{jt}\right)^2\right\}^2$$
$$= N_k b_{kk} - \sum_{l=1}^{K} N_l[b_{kl}(1-b_{k'l})^2 + b_{k'l}^2(1-b_{kl})]^2$$
$$+ O(N_{k'} b_{k',k'}^4 + b_{kk} + \sum_{l\neq k} N_l b_{kl})$$
$$= N_k b_{kk} + O\left(\sum_{l=1}^{K} N_l b_{kl}^2\right)$$

A.3 Proof of Theorem 2.1

We now focus on deriving lower bound for $P(\mathcal{E})$. Recall that $\delta_{i,k} = \|Z_i - \widehat{\alpha}_k\|^2 = \sum_{s=1}^{d} X_s^\top \Omega_{i,k} X_s$, where $\Omega_{i,k} = (A_i - N_k^{-1}\sum_{j:C_j=k} A_j)(A_i - N_k^{-1}\sum_{j:C_j=k} A_j)^\top$. Define event $\mathcal{E}_0(\eta) = \{\max_{i,k:C_i=k} |\delta_{i,k} - E(\delta_{i,k})| \le \eta\}$ and $\mathcal{E}_1(\eta) = \{\max_{i,k:C_i\neq k} |\delta_{i,k} - E(\delta_{i,k})| \le \eta\}$. Recall that the difference between the expectations of $\delta_{i,k'}$ $(C_i = k)$ for $k' = k$ and $k' \neq k$ is $N_k b_{kk}^2 + N_{k'} b_{k',k'}^2 + O(\sum_{l\neq k} N_l b_{kl}^2)$. If we pick a sufficiently small $\eta = (N_k b_{kk}^2 + N_{k'} b_{k',k'}^2)/2$, then

$$(6)\quad P(\mathcal{E}) \ge P\{\mathcal{E}_0(\eta)\cup\mathcal{E}_1(\eta)\} \ge 1 - P\{\mathcal{E}_0^c(\eta)\} - P\{\mathcal{E}_1^c(\eta)\},$$

For a given $k \in \{1,2,..,K\}$, we further define $\mathcal{E}_0(k,\eta) = \{\max_{i:C_i=k} |\delta_{i,k} - E(\delta_{i,k})| \le \eta\}$ and $\mathcal{E}_1(k,\eta) = \{\max_{C_i\neq k} |\delta_{i,k} - E(\delta_{i,k})| \le \eta\}$. Then we have

$$(7)\ P(\mathcal{E}_0^c(\eta)) = P\{\cup_{k=1}^{K}\mathcal{E}_0^c(k,\eta)\} \le \sum_{k=1}^{K} P(\mathcal{E}_0^c(k,\eta)),$$
$$(8)\ P(\mathcal{E}_1^c(\eta)) = P\{\cup_{k=1}^{K}\mathcal{E}_1^c(k,\eta)\} \le \sum_{k=1}^{K} P(\mathcal{E}_1^c(k,\eta)),$$

Thus it suffices to show $P\{\mathcal{E}_0^c(k,\eta)\}$ and $P\{\mathcal{E}_1^c(k,\eta)\}$, respectively. We derive $P\{\mathcal{E}_0^c(k,\eta)\}$ first and $P\{\mathcal{E}_1^c(k,\eta)\}$ can be obtained similarly.

Note that $\delta_{i,k}$ contains two parts of randomness, which are from the adjacency matrix A and random projection matrix X, respectively. We define the event $\mathcal{A} = \{c_1 n\theta_n \le |\mathrm{tr}(\Omega_{i,k})| \le c_2 n\theta_n\}$ and show the following two inequalities

$$(9)\qquad P(\mathcal{A}^c) \le 2\exp(-c_3 n\theta_n),$$
$$(10) P\left(\left\{\left|\delta_{i,k} - E(\delta_{i,k})\right| > \eta\right\}\middle| A, \mathcal{A}\right) \le 2\exp\left(-c_4 d\theta_n^2\right),$$

where c_1, c_2, c_3 and c_4 are some positive constants. Combine the results in (9) and (10), we can derive that

$$P\left(\left|\delta_{i,k} - E(\delta_{i,k})\right| > \eta\right)$$
$$= P\left(\left\{\left|\delta_{i,k} - E(\delta_{i,k})\right| > \eta\right\}\cap\mathcal{A}\right)$$
$$+ P\left(\left\{\left|\delta_{i,k} - E(\delta_{i,k})\right| > \eta\right\}\cap\mathcal{A}^c\right)$$
$$\le 2\exp\left(-c_3 n\theta_n\right) + 2\exp\left(-c_4 d\theta_n^2\right)$$

It yields,

$$P\{\mathcal{E}_0^c(k,\eta)\} = P\left(\max_{C_i=k}|\delta_{i,k} - E(\delta_{i,k})| > \eta\right)$$
$$\le \sum_{C_i=k} P\left(|\delta_{i,k} - E(\delta_{i,k})| > \eta\right)$$
$$(11)\quad \le 2N_k\exp\left(-c_3 n\theta_n\right) + 2N_k\exp\left(-c_4 d\theta_n^2\right).$$

In the following we derive (9) and (10) in the following two parts and then state the result for $P\{\mathcal{E}_0^c(k,\eta)\}$.

1. Derivation of (9)

Note that $\mathrm{tr}(\Omega_{i,k})$ can be represented into independent summations as $\mathrm{tr}(\Omega_{i,k}) = \sum_{t=1}^{N} W_t$, where $W_t = (a_{it} - N_k^{-1}\sum_{j:C_j=k} a_{jt})^2 \le 1$ for $1 \le t \le N$ are bounded and independent with each others. Then by Bernstein's inequality, we have

$$P\left\{|\mathrm{tr}(\Omega_{i,k}) - E[\mathrm{tr}(\Omega_{i,k})]| > \epsilon\right\}$$
$$(12)\quad \le 2\exp\left(-\frac{\epsilon^2/2}{\mathrm{var}[\mathrm{tr}(\Omega_{i,k})] + \epsilon/3}\right)$$

Recall that the leading term of $E[\mathrm{tr}(\Omega_{i,k})]$ and $\mathrm{var}[\mathrm{tr}(\Omega_{i,k})]$ are both $N_k b_{kk}$ by (3) and (5), which is of order $O(n\theta_n)$ by

assumptions. By choosing appropriate ϵ of order $O(n\theta_n)$, we know that there exists three positive constant c_1, c_2 and c_3 such that

$$P\left\{c_1 n\theta_n \le \left|\operatorname{tr}(\Omega_{i,k})\right| \le c_2 n\theta_n\right\}$$
$$\ge 1 - 2\exp\left(-c_3 n\theta_n\right) \tag{13}$$

Let $\mathcal{A} = \left\{c_1 n\theta_n \le \left|\operatorname{tr}(\Omega_{i,k})\right| \le c_2 n\theta_n\right\}$, then (13) leads to (9).

2. Derivation of (10)

Conditional on $\mathcal{A}$ and adjacency matrix A, $\Omega_{i,k}$ can be rewritten into a spectral decomposition form as $\Omega_{i,k} = \operatorname{tr}(\Omega_{i,k})u_i u_i^\top$, where $\|u_i\| = 1$. Then we have $X_s^\top \Omega_{i,k} X_s = \operatorname{tr}(\Omega_{i,k})(X_s^\top u_i)^2 \sim \operatorname{tr}(\Omega_{i,k})\chi^2(1)$. Let $Y_s = (n\theta_n)^{-1}\operatorname{tr}(\Omega_{i,k})(X_s^\top u_i)^2$ and then $\delta_{i,k} = n\theta_n \sum_{s=1}^d Y_s$. We can verify that

$$\mathbb{E}(|Y_s^l||\mathcal{A}, A) = \{\operatorname{tr}(\Omega_{i,k})/(n\theta_n)\}^l (2l-1)!!$$
$$\le 2\{\operatorname{tr}(\Omega_{i,k})/(n\theta_n)\}^2 (2c_2)^{l-2} l!$$

Therefore, given A and $\mathcal{A}$, by Bernstein's inequality [34] we have for any $\epsilon > 0$

$$P\left(\left\{\left|\sum_{k=1}^d Y_s - E\sum_{s=1}^d Y_s\right| > \frac{\epsilon}{n\theta_n}\right\}\right)$$
$$\le 2\exp\left\{-\frac{\epsilon^2}{2n^2\epsilon_n^2\sum_{s=1}^d \operatorname{var}(Y_s|\mathcal{A}, A) + c_2 n\theta_n \epsilon}\right\}$$
$$\le 2\exp\left\{-\frac{\epsilon^2}{4dn^2\theta_n^2 c_2^2 + 2c_2 n\theta_n \epsilon}\right\}$$

Choose $\epsilon = \eta = d(N_k b_{kk}^2 + N_{k'} b_{k',k'}^2)/2$, then there exists a constant c_4 such that

$$P\left(\left\{\left|\delta_{i,k} - E(\delta_{i,k})\right| > \eta\right\}\middle| A, \mathcal{A}\right) \le 2\exp\left(-c_4 d\theta_n^2\right)$$

As a result, we have

$$P(\mathcal{E}_0^c(\eta)) \le \sum_{k=1}^K P(\mathcal{E}_0^c(k,\eta))$$
$$\le 2N\exp\left(-c_3 n\theta_n\right) + 2N\exp\left(-c_4 d\theta_n^2\right) \tag{14}$$

The calculation of $P(\mathcal{E}_1^c)$ is similar. Recall that the leading term of $E[\operatorname{tr}(\Omega_{i,k})]$ and $\operatorname{var}[\operatorname{tr}(\Omega_{i,k})]$ are both $N_{k'} b_{k',k'}$ for $C_i = k' \ne k$, which is of order $O(n\theta_n)$. Following the same argument as $C_i = k$, we can show

$$P\left(\left|\delta_{i,k} - E(\delta_{i,k})\right| > \eta\right) \le 2\exp\left(-c_3 n\theta_n\right) + 2\exp\left(-c_4 d\theta_n^2\right).$$

The details are omitted here due to the duplication. Then we can verify that

$$P\{\mathcal{E}_1^c(k,\eta)\} = P\left(\max_{i:C_i\ne k} |\delta_{i,k} - E(\delta_{i,k})| > \eta\right)$$
$$\le \sum_{i:C_i\ne k} P\left(|\delta_{i,k} - E(\delta_{i,k})| > \eta\right)$$
$$\le 2(N - N_k)\exp(-c_3 n\theta_n) + 2(N - N_k)\exp(-c_4 d\theta_n^2) \tag{15}$$

As a result, we can derive

$$P(\mathcal{E}_1^c(\eta)) \le \sum_{k=1}^K P(\mathcal{E}_1^c(k,\eta))$$
$$\le 2(K-1)N\exp\left(-c_3 n\theta_n\right) + 2(K-1)N\exp\left(-c_4 d\theta_n^2\right) \tag{16}$$

Combine the results in inequality (14) and (16), then we have

$$P(\mathcal{E}) \ge P\{\mathcal{E}_0(\eta)\cup\mathcal{E}_1(\eta)\} \ge 1 - P\{\mathcal{E}_0^c(\eta)\} - P\{\mathcal{E}_1^c(\eta)\}$$
$$\ge 1 - 2\exp\left(-C\min\left\{n\theta_n, d\theta_n^2\right\} + \log K + \log N\right),$$

for some positive constant C. This completes the proof of Theorem 1.

ACKNOWLEDGEMENTS

The authors thank the editor, guest editor, the associate editor and anonymous referees for constructive suggestions. Haobo Qi and Hansheng Wang are supported by National Natural Science Foundation of China (NSFC, 11831008). Xuening Zhu's research is supported by the National Natural Science Foundation of China (nos. 72222009, 11901105, 71991472, U1811461), and the Shanghai Sailing Program for Youth Science and Technology Excellence (19YF1402700).

Received 5 December 2021

REFERENCES

[1] Abbe, E. (2018). Community detection and stochastic block models. *Foundations & Trends in Communications & Information Theory* **14**. MR3827065

[2] Adamic, L. A. and Glance, N. (2005). The political blogosphere and the 2004 U.S. election: divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery* 36–43. Association for Computing Machinery.

[3] 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. MR3127859

[4] Arthur, D. and Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *SODA '07*. MR2485254

[5] Bickel, P. J. and Chen, A. (2009). A nonparametric view of network models and Newman-Girvan and other modularities. *Proceedings of the National Academy of Sciences* **106** 21068–21073.

[6] Bickel, P. J. and Sarkar, P. (2016). Hypothesis testing for automated community detection in networks. *Journal of the Royal Statistical Society. Series B: Statistical Methodology* **78** 253–273. MR3453655

[7] Clauset, A., Newman, M. and Moore, C. (2006). Finding community structure in very large networks. *Physical Review E* **70** 066111.

[8] De, Meo, P., Ferrara, E., Provetti, A., Fiumara and G. (2014). Mixing local and global information for community detection in large networks (Conference Paper). *Journal of Computer and System Sciences*. MR3105909
[9] Gao, C., Ma, Z., Zhang, A. Y. and Zhou, H. H. (2018). Community detection in degree-corrected block models. *The Annals of Statistics* **46** 2153–2185. MR3845014
[10] Handcock, M. S., Raftery, A. E. and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society. Series A: Statistics in Society* **170** 301–354. MR2364300
[11] Holland, P. W., Laskey, K. B. and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks* **5** 109–137. MR0718088
[12] Hu, J., Qin, H., Yan, T. and Zhao, Y. (2020). Corrected Bayesian information criterion for stochastic block models. *Journal of the American Statistical Association* **115** 1771–1783. MR4189756
[13] Huang, D., Yin, J., Shi, T. and Wang, H. (2016). A statistical model for social network labeling. *Journal of Business & Economic Statistics* **34** 368–374. MR3523781
[14] Härdle, W. K., Wang, W. and Yu, L. (2016). TENET: Tail-Event driven NETwork risk. *Journal of Econometrics* **192** 499–513. Innovations in Multiple Time Series Analysis. MR3488092
[15] Jing, L. (2014). A Goodness-of-fit test for stochastic block models. *The Annals of Statistics* **44**. MR3449773
[16] Joseph, A. and Yu, B. (2016). Impact of regularization on spectral clustering. *Annals of Statistics* **44** 1765–1791. MR3519940
[17] Karrer, B. and Newman, M. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E* **83** 016107. MR2788206
[18] Lawrence, H. and Phipps, A. (1985). Comparing partitions. *Journal of Classification* **2** 193–185.
[19] Le, C. M. and Levina, E. (2015). Estimating the number of communities in networks by spectral methods. *ArXiv* **abs/1507.00827**. MR4422967
[20] Lei, J. and Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics* **43** 215–237. MR3285605
[21] Li, T., Levina, E. and Zhu, J. (2020). Network cross-validation by edge sampling. *Biometrika* **107** 257–276. MR4108931
[22] Liu, X., Patacchini, E. and Rainone, E. (2017). Peer effects in bed time decisions among adolescents: A social network model with sampled data. *The Econometrics Journal* **20** S103–S125.
[23] Luxburg, U. V. (2007). A Tutorial on Spectral Clustering. *Statistics and Computing* **17** 395–416. MR2409803
[24] Marbach, D., Holmes, B. R., Kellis, M. and Consortium (2012). Wisdom of crowds for robust gene network inference. *Nature Methods* **9** 796–804.
[25] Marbach, D., J., P. R., T., S., C., M., Floreano, D. and Stolovitzky, G. (2010). Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences* **107** 6286–6291.
[26] Newman, M. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E* **69** 026113.
[27] Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E* **69** 066133.
[28] Ovelgönne, M. and Geyer-Schulz, A. (2013). *An ensemble learning strategy for graph clustering*. An ensemble learning strategy for graph clustering. MR3074905
[29] Qin, T. and Rohe, K. (2013). Regularized spectral clustering under the degree-corrected stochastic block model. *advances in neural information processing systems*.
[30] Riedy, J., Bader, D. A. and Meyerhenke, H. (2012). Scalable multi-threaded community detection in social networks. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops PhD Forum* 1619–1628.
[31] Rohe, K., Chatterjee, S. and Yu, B. (2010). Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics* **39** 1878–1915. MR2893856
[32] Sojourner, A. (2013). Identification of peer effects with missing peer data: evidence from project STAR. *Economic Journal* **123**.
[33] Su, L., Wang, W. and Zhang, Y. (2017). Strong consistency of spectral clustering for stochastic block models. *Economics and Statistics Working Papers*.
[34] Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press. MR3967104
[35] Zhao, Y., Levina, E. and Zhu, J. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics* **40** 2266–2292. 0 MR3059083

Haobo Qi
Guanghua School of Management
Peking University
Beijing
P.R.China
E-mail address: qihaobo_gsm@pku.edu.cn

Xuening Zhu
School of Data Science
Fudan University
Shanghai
P.R.China
E-mail address: xueningzhu@fudan.edu.cn

Hansheng Wang
Guanghua School of Management
Peking University
Beijing
P.R.China
E-mail address: hansheng@pku.edu.cn