Statistica Sinica

COMMUNITY EXTRACTION OF NETWORK DATA UNDER STOCHASTIC BLOCK MODELS

Quan Yuan[1], Binghui Liu[1,*], Danning Li[1,*], Yanyuan Ma[2]

[1] *Northeast Normal University and* [2] *Pennsylvania State University*

Abstract: Most existing community discovery methods focus on partitioning all nodes of the network into communities. However, many real networks contain background nodes that do not belong to any community. In such a situation, typical methods tend to artificially split the background nodes and group them together with communities with relatively stronger connection, hence lead to distorted results. To avoid this, some community extraction methods have been developed to achieve community discovery with background nodes, which are based on searching algorithms, hence have difficulties in handling large-scale networks due to high computational complexity. To this end, in this paper we propose some algorithms with polynomial complexity to achieve community extraction of large-scale networks. We rigorously show that the proposed algorithms have attractive theoretical properties. In particular, the estimators of the community labels using the proposed algorithms reaches the asymptotic minimax risk under the community extraction model, a specific stochastic block model. Then, we illustrate the advantages and feasibility of the proposed algorithms via extensive

*Corresponding authors.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0372

simulated networks and a political blog network.

Key words and phrases: background nodes, community extraction, refinement algorithm.

1. Introduction

Networks are widely used to represent and analyze the relationship between interacting units in complex systems (Goldenberg et al., 2010; Wasserman and Faust, 1994). In network data analysis, community discovery is a fundamental problem, which aims to divide the nodes of the network into communities, so that the nodes in the same community are closely connected, while the nodes from different communities are loosely connected. Identifying communities can provide important insights about network organizations. There is a large number of literature on community discovery from different research fields, such as computer science (Flake et al., 2002), social science (Moody and White, 2003) and genetics (Spirin and Mirny, 2003). We refer to Fortunato (2010), Fortunato and Hric (2016) and Zhao (2017) for comprehensive reviews on this topic.

Most literatures on community discovery study the problem without "background nodes", where the background nodes are defined as the weakly connected nodes that have no strong association with any community of

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0372

the network. However, there are indeed many examples where background nodes exist (Zhao et al., 2011). Applied to networks with background nodes, typical community discovery methods tend to split up weakly connected nodes and group them together with tighter communities. To better handle such situation, community discovery with background nodes began to receive much attention, which was specially named "community extraction", aiming to recover the communities and extract the background nodes at the same time (Zhao et al., 2011; Wilson et al., 2017).

Like community discovery, community extraction is also a computationally challenging problem in large-scale networks, because the number of possible partitions of nodes into non-overlapping groups is non-polynomial in the size of a network. For typical community discovery, a huge number of algorithmic approaches have been proposed (Fortunato, 2010), including many heuristic algorithms, such as normalized cuts (Shi and Malik, 2000), modularity optimization (Newman and Girvan, 2004), spectral methods (Lei and Rinaldo, 2015) and non-negative matrix factorization (Wang et al., 2011), to name just a few. In addition, many statistical approaches have been proposed based on some probabilistic models (Amini et al., 2013; Wang et al., 2020), such as the stochastic block model (SBM) (Holland et al., 1983) and degree-corrected stochastic block model (DCSBM)

(Karrer and Newman, 2011).

In contrast, there are much fewer studies on community extraction. The problem of community extraction was originally studied by Zhao et al. (2011). They established a community extraction criterion based on ratio cut (Wei and Cheng, 1989) and proposed a heuristic algorithm, i.e. a tabu search algorithm, to maximize the extraction criterion over all possible choices. Then, they derived the asymptotic consistency of the one-step maximizer of the extraction criterion under a community extraction model based on SBM. Later, Wilson et al. (2017) extended community extraction to multi-layer networks and proposed a community extraction method by maximizing the multi-layer extraction score based on modularity. They considered the asymptotic consistency of the maximizer of the multi-layer extraction score based on a multi-layer SBM.

These methods are highly instructive and useful in small-scale networks, but unfortunately they are based on searching algorithms, hence are not computationally efficient to deal with large-scale networks. On this ground, in this paper, we propose some fast algorithms with polynomial complexity to achieve community extraction of large-scale networks, and rigorously show that the proposed algorithms have attractive theoretical properties. In particular, the estimators of the community labels using some of the

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0372

proposed algorithms reaches the asymptotic minimax risk under the community extraction model, a specific stochastic block model.

Specifically, we proposed a two-step refinement algorithm for community extraction, and present that under certain conditions the proposed refinement algorithm initialized by two algorithms based on low rank approximation and spectral clustering, respectively, reaches the established asymptotic minimax risk under the community extraction model. Hence, the asymptotic minimax risk specially for community extraction is first established in this paper. Then, we illustrate the advantages of the proposed algorithms for community extraction via extensive simulation studies and a practical application.

Our study on community extraction has conquered new challenges in both algorithm and theory. First, existing methods often struggle to find a suitable initialization method based on spectral clustering to handle networks with background nodes. This is because in a network model with background nodes, the signal-to-noise ratio of the Kth eigenvector of the adjacency matrix may be significantly low, where K is the number of clusters of network nodes, including $K-1$ communities and a set of background nodes. To deal with the first challenge, we propose an initialization method that is more suitable for identifying background node set considering the signal-to-noise ratio problem. Second, in the refinement step of many existing two-step methods, the fundamental concept is to assign a node to the cluster with which it has the closest connection. However, this approach may not effectively handle background nodes. Hence, we propose a refinement based on the likelihood information of the community extraction model that we studied. Finally, when establishing the upper or lower bounds of the asymptotic minimax risk in the network model including background nodes, many conditions used in existing studies are somewhat unreasonable and restrictive. This is described in terms of the upper and lower bounds of the asymptotic minimax risk respectively in Section $S.5.3$ of the Supplement. Hence, we use new tools to establish the asymptotic minimax risk under more relaxed conditions.

The rest of this paper is organized as follows. In Section 2, we propose a refinement algorithm and their initialization algorithms for community extraction. In Section 3, we establish the theoretical results of the proposed algorithms. Then, we compare the proposed refinement algorithm with some of its competitors via extensive simulation results in Section 4, followed by a practical application in Section 5. We conclude this paper in Section 6, and relegate the technical proofs as well as some additional simulation results and discussions to the Supplement.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0372

2. Algorithms for community extraction

In this section, we will propose some algorithms for community extraction.

2.1 Notation

First, we present some general definitions and notation. For any positive integer n, let $[n] = \{1, \cdots, n\}$. For any set $\mathcal{S}$, let $|\mathcal{S}|$ denote the number of its elements. For a positive real number x, let $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the largest integer not greater than x and the smallest integer not less than x, respectively. For two positive sequences $\{x_n\}_{n=1}^{\infty}$ and $\{y_n\}_{n=1}^{\infty}$, $x_n \gtrsim y_n$ means that $x_n \geqslant C y_n$ for some constant $C > 0$; $x_n \lesssim y_n$ means that $x_n \leqslant C y_n$ for some constant $C > 0$; $x_n \asymp y_n$ means that $\frac{1}{C} y_n \leqslant x_n \leqslant C y_n$ for some constant $C \geqslant 1$; $x_n \gg y_n$ means that $y_n = o(x_n)$; $x_n \ll y_n$ means $x_n = o(y_n)$. For a vector $\boldsymbol{x} = (x_1, \cdots, x_n)^\top \in \mathbb{R}^n$, $\|\boldsymbol{x}\|_2 = \sqrt{\sum x_i^2}$. For a matrix $\boldsymbol{M} = (M_{ij})_{n \times n} \in \mathbb{R}^{n \times n}$, $\|\boldsymbol{M}\|_{\mathrm{F}} = (\sum_{i=1}^n \sum_{j=1}^n M_{ij}^2)^{1/2}$ and $\|\boldsymbol{M}\|_{\mathrm{op}} = s_{\max}(\boldsymbol{M})$, where $s_{\max}(\boldsymbol{M})$ denotes the largest singular value of $\boldsymbol{M}$. For two matrices $\boldsymbol{A} = (A_{ij})_{m \times n}$ and $\boldsymbol{B} = (B_{ij})_{m \times n} \in \mathbb{R}^{m,n}$, $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{i=1}^m \sum_{j=1}^n A_{ij} B_{ij}$. Note that in this paper we will use η to represent a sequence that tends to 0 in some places later, which represents different sequences in different places.

2.2 SBM for community extraction

We consider the undirected network $G = (V, E)$ with node set $V = [n]$ and edge set $E \subseteq \{(i, j) : i, j \in V\}$, which can be formulated by the adjacency matrix $\boldsymbol{A} = (A_{ij})_{n\times n} \in \{0, 1\}^{n\times n}$. Here, $A_{ij} = 1$ if $(i, j) \in E$, otherwise $A_{ij} = 0$. Suppose that there is no self-loop in network G, i.e. $A_{ii} = 0$ for each node $i \in V$.

For community extraction of network G, the nodes of G can be divided into two categories: community nodes and background nodes. Specifically, each community node belongs to one community of G, which generally has more connections with nodes belonging to the same community than with nodes belonging to different communities or outside all communities. On the contrary, a background node does not belong to any community of G, which has relatively few connections to all communities. Suppose that network G has $K - 1$ communities and some background nodes. Let $\boldsymbol{c} = (\boldsymbol{c}(1), \cdots, \boldsymbol{c}(n))^{\top}$ denote the community label vector of network G, where for each $i \in [n]$, we let $\boldsymbol{c}(i) = K$ if i is a background node, otherwise let $\boldsymbol{c}(i) = k$ if i belongs to community k for $k \in [K - 1]$.

The network G with both community nodes and background nodes can be characterized by the following SBM:

$$\begin{aligned} A_{ij} = A_{ji} \overset{ind}{\sim} \text{Bern}\left(P_{\boldsymbol{c}(i)\boldsymbol{c}(j)}\right) \text{ for all } i < j \in [n], \\ A_{ii} \equiv 0 \text{ for all } i \in [n], \qquad (2.1) \\ \min_{k \neq K} P_{kk} > \max_{\substack{u \neq v \\ u,v \in [K-1]}} P_{uv},\ \min_{k \neq K} P_{kk} > P_{Kl} \equiv q \text{ for all } l \in [K], \end{aligned}$$

where the edge-probability matrix $\boldsymbol{P} = (P_{kl})_{K \times K} \in [0,1]^{K \times K}$ with $P_{kl} \equiv P_{lk}$ and the community label vector $\boldsymbol{c} = (\boldsymbol{c}(1), \cdots, \boldsymbol{c}(n))^\top \in [K]^n$ are model parameters. The constraints $\min_{k \neq K} P_{kk} > P_{Kl} \equiv q$ are imposed on $\boldsymbol{P}$ to ensure that a background node is connected to any other node with a very low probability, and there is no difference in the connection probability between a background node and any other node.

Under SBM, the premise that communities can be detected is that there exists some type of separability between communities in terms of the probability matrix $\boldsymbol{P}$. For example, Yun and Proutiere (2016) defined this separability as the separability between any two rows of $\boldsymbol{P}$. In fact, the proposed model in (2.1) and the model studied in Zhang and Zhou (2016) exhibit such type of separability. Note that both Yun and Proutiere (2016) and Zhang and Zhou (2016) require that the node numbers of different communities or groups are of the same order, whereas in this paper, our study includes situations that allow for significant differences between the node numbers of different groups. In addition, Yun and Proutiere (2016) requires that the values in each row of $\boldsymbol{P}$ are of the same order, but such requirement is not needed in our study.

2.3 Refinement algorithms for community extraction

In this subsection, we first propose a two-step Refinement Algorithm for Community Extraction, abbreviated as RACEn and partially inspired by the algorithm frameworks of Gao et al. (2017) and Gao et al. (2018), where the symbol n next to RACE means that it need to apply the initialization algorithm for n times. For each $i \in [n]$, let $\boldsymbol{A}_{-i} \in \{0,1\}^{(n-1)\times(n-1)}$ denote the submatrix of the adjacency matrix $\boldsymbol{A}$, which is obtained by removing the ith row and column of $\boldsymbol{A}$.

Algorithm 1. (**RACEn**)

Input: *The adjacency matrix* $\boldsymbol{A} \in \{0,1\}^{n\times n}$, *an initialization algorithm and the specific value of* $K \geqslant 2$.

Output: *An estimator* $\check{\boldsymbol{c}} \in [K]^n$ *of the community label vector* $\boldsymbol{c} \in [K]^n$.

1. *(Initialization) For each* $i \in [n]$, *applying the initialization algorithm to* $\boldsymbol{A}_{-i}$, *we get the output* $(\boldsymbol{c}^{0i}(1), \ldots, \boldsymbol{c}^{0i}(i-1), \boldsymbol{c}^{0i}(i+1), \ldots, \boldsymbol{c}^{0i}(n))^\top$, *which is a vector with length* $n-1$. *Define* $\boldsymbol{c}^{0i} = (\boldsymbol{c}^{0i}(1), \ldots, \boldsymbol{c}^{0i}(n))^\top$ *with* $\boldsymbol{c}^{0i}(i) = 0$.

2. *(Refinement) For each $k, l \in [K]$, let*

$$P^{0i}_{kl} = \begin{cases} \dfrac{\sum_{u<v} A_{uv}\mathbb{I}\{\boldsymbol{c}^{0i}(u)=k, \boldsymbol{c}^{0i}(v)=k\}}{\frac{1}{2}n^{0i}_k(n^{0i}_k-1)}, & k = l, \\ \dfrac{\sum_{u,v\in[n]} A_{uv}\mathbb{I}\{\boldsymbol{c}^{0i}(u)=k, \boldsymbol{c}^{0i}(v)=l\}}{n^{0i}_k n^{0i}_l}, & k \neq l, \end{cases}$$

with $n^{0i}_k = \sum_{j\in[n]} \mathbb{I}\{\boldsymbol{c}^{0i}(j) = k\}$*. Let* $\hat{q}^{0i} = \{\sum_{k=1}^{K-1} n^{0i}_k P^{0i}_{Kk} + (n^{0i}_K - 1)P^{0i}_{KK}/2\}/\{\sum_{k=1}^{K-1} n^{0i}_k + (n^{0i}_K - 1)/2\}$*, and update* $P^{0i}_{Kl} = \hat{q}^{0i}$ *for all* $l \in [K]$*. For each* $i \in [n]$*, let*

$$\check{\boldsymbol{c}}(i) = \underset{k\in[K]}{\operatorname{argmax}} \sum_{l=1}^{K-1} \sum_{j:\boldsymbol{c}^{0i}(j)=l} \Big\{A_{ij} \log P^{0i}_{kl} + (1 - A_{ij}) \log(1 - P^{0i}_{kl})\Big\}.$$

Assuming that the initial estimator is reasonable and reliable, then for node i, the refinement step can be viewed as a majority voting decision based on the initial result $\boldsymbol{c}^{0i}$. If $\boldsymbol{c}^{0i}(j) = \boldsymbol{c}(j)$ for each $j \neq i$, then $\check{\boldsymbol{c}}(i)$ is the MLE of $\boldsymbol{c}(i)$.

Obviously, the performance of Algorithm 1 critically depends on the properties of the initialization algorithm. The output of the initialization algorithm needs to perform reasonably well. Next, we show that the initial estimator only needs to satisfy a certain weak consistency criterion, based on which the refinement step of Algorithm 1 will lead to an output with optimal misclassification proportion.

Similar to Gao et al. (2018), applying the initialization algorithm for n times in Algorithm 1 can facilitate the technical proof for establishing the theoretical results of the refinement algorithm. However, when n is huge, repeating the initialization algorithm for n times may be very time-consuming. Therefore, in practical applications, we usually use its accelerated version Algorithm S1, abbreviated as RACE, to replace it. We relegate Algorithm S1 to the Supplement. It is worth mentioning that RACE only runs the initialization algorithm once, thus accelerating the speed. In fact, suggested by some numerical results presented in Section S.1.1, the community extraction performance of Algorithm 1 and Algorithm S1 is extremely similar. Hence, in the following simulation study and real data analysis, we will all use RACE.

Note that the performance of the proposed refinement algorithm may largely rely on good performance of the initialization algorithm, which we will study next.

2.4 Initialization algorithm

We propose an initialization algorithm, abbreviated as INIT, based on low rank approximation, for Algorithms 1 and S1, similar to the initialization algorithm proposed by Gao et al. (2018) for community discovery without background nodes, whose output satisfies Condition 1 in Theorem 3.

Algorithm 2. (**INIT**)

Input: *The adjacency matrix* $\boldsymbol{A} \in \{0,1\}^{n\times n}$*, the specific value of* K*, the threshold parameter* τ *and the approximation parameter* ξ*.*

Output: *An estimator* $\boldsymbol{c}_{\mathit{init}}^0 = (\boldsymbol{c}_{\mathit{init}}^0(1), \cdots, \boldsymbol{c}_{\mathit{init}}^0(n))^\top \in [K]^n$ *of the community label vector* $\boldsymbol{c} \in [K]^n$*.*

1. *Define* $\boldsymbol{A}^\tau \in \{0,1\}^{n\times n}$ *by replacing all elements in the* i*-th row and column of* $\boldsymbol{A}$ *with zero, if the sum of the* i*-th row of* $\boldsymbol{A}$ *is larger than* τ*, for each* $i \in [n]$*.*

2. *Solve the following low rank approximation problem*

$$\tilde{\boldsymbol{M}} = \mathop{\arg\min}_{\substack{\mathrm{rank(M)} \leqslant \mathrm{K} \\ \mathrm{M} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}}} \|\boldsymbol{A}^\tau - \boldsymbol{M}\|_{\mathrm{F}}^2 \tag{2.2}$$

by singular value decomposition.

3. *For each* $i \in [n]$*, let* $\tilde{\boldsymbol{M}}_i$ *denote the transpose of the ith row of* $\tilde{\boldsymbol{M}}$*. Solve the following* $(1+\xi)$*-approximation* K*-means optimization problem: find some* $\tilde{\boldsymbol{c}}_{\mathit{init}}^0 = (\tilde{\boldsymbol{c}}_{\mathit{init}}^0(1), \cdots, \tilde{\boldsymbol{c}}_{\mathit{init}}^0(n))^\top \in [K]^n$*, such that*

$$\sum_{k=1}^K \min_{\boldsymbol{\nu}_k \in \mathbb{R}^n} \sum_{i:\tilde{\boldsymbol{c}}_{\mathit{init}}^0(i)=k} \|\tilde{\boldsymbol{M}}_i - \boldsymbol{\nu}_k\|_2^2 \leqslant (1+\xi) \min_{\boldsymbol{c}\in[K]^n} \sum_{k=1}^K \min_{\boldsymbol{\nu}_k \in \mathbb{R}^n} \sum_{i:\boldsymbol{c}(i)=k} \|\tilde{\boldsymbol{M}}_i - \boldsymbol{\nu}_k\|_2^2. \tag{2.3}$$

4. *For each* $k, l \in [K]$*, let*

$$\tilde{P}_{kl}^0 = \begin{cases} \dfrac{\sum\limits_{i<j} A_{ij}^\tau \mathbb{I}\{\tilde{\boldsymbol{c}}_{\mathit{init}}^0(i)=k, \tilde{\boldsymbol{c}}_{\mathit{init}}^0(j)=k\}}{\frac{1}{2}\tilde{n}_k^0(\tilde{n}_k^0-1)}, & k = l, \\ \dfrac{\sum\limits_{i,j\in[n]} A_{ij}^\tau \mathbb{I}\{\tilde{\boldsymbol{c}}_{\mathit{init}}^0(i)=k, \tilde{\boldsymbol{c}}_{\mathit{init}}^0(j)=l\}}{\tilde{n}_k^0 \tilde{n}_l^0}, & k \neq l, \end{cases}$$

with $\tilde{n}_k^0 = \sum_{j\in[n]} \mathbb{I}\{\tilde{\boldsymbol{c}}_{\mathit{init}}^0(j) = k\}$*. Let*

$$k^* = \arg\min_{k\in[K]} \tilde{P}_{kk}^0. \tag{2.4}$$

Then, for each i *with* $\tilde{\boldsymbol{c}}_{\mathit{init}}^0(i) = k^*$*, let* $\boldsymbol{c}_{\mathit{init}}^0(i) = K$*; for each* i *with* $\tilde{\boldsymbol{c}}_{\mathit{init}}^0(i) = K$*, let* $\boldsymbol{c}_{\mathit{init}}^0(i) = k^*$*; and let* $\boldsymbol{c}_{\mathit{init}}^0(i) = \tilde{\boldsymbol{c}}_{\mathit{init}}^0(i)$*, for each remaining node.*

Note that in the first step of Algorithm 2, the elements of the rows and columns of $\boldsymbol{A}$ whose sums are too large are replaced with zero to improve the denoising effect of the proposed algorithm in sparse regime. Such strategy was previously used in Chin et al. (2015) and Gao et al. (2018). If this strategy is not adopted, the high probability error bound for the output of Algorithm 2 would suffer an extra multiplier of order $O(\log n)$.

Note that to solve the $(1 + \xi)$-approximation K-means optimization problem in step 3 of Algorithm 2, some methods have been studied, such as the $(1 + \xi)$-approximation algorithm proposed by Kumar et al. (2004). However, such methods are mainly used for theoretical investigation rather than practical implementation. Hence, in the later simulation studies and real data analysis, we will use the classical K-means algorithm in Hartingan and Wong (1979) to replace a $(1 + \xi)$-approximation algorithm in step 3 of Algorithm 2, to approximately solve the $(1 + \xi)$-approximation K-means optimization problem.

Note that Algorithm 2 (INIT) has some drawbacks: in its K-means clustering step, it needs to cluster n n-dimensional vectors, which makes it very time-consuming, especially when n is particularly large. Besides, it will have poor performance when the signal is relatively small. To this end, we propose using the following spectral clustering algorithm, abbreviated as ESC, as the initialization algorithm, which is evolved from the spectral clustering algorithm proposed by Lei and Rinaldo (2015) but more suitable for dealing with networks with background nodes.

Algorithm 3. (**ESC**)

Input: *The adjacency matrix* $\boldsymbol{A} \in \{0,1\}^{n\times n}$*, the specific value of* K*, the threshold parameter* τ *and the approximation parameter* ξ*.*

Output: *An estimator* $\boldsymbol{c}^0_{\mathsf{esc}} = (\boldsymbol{c}^0_{\mathsf{esc}}(1), \cdots, \boldsymbol{c}^0_{\mathsf{esc}}(n))^\top \in [K]^n$ *of the label vector* $\boldsymbol{c} \in [K]^n$*.*

1. *Define* $\boldsymbol{A}^\tau \in \{0,1\}^{n\times n}$ *by replacing all elements in the ith row and column of* $\boldsymbol{A}$ *with zero, if the sum of the ith row of* $\boldsymbol{A}$ *is larger than* τ*, for each* $i \in [n]$*.*

2. *Calculate* $\hat{\boldsymbol{U}}^{K-1} \in \mathbb{R}^{n\times(K-1)}$ *consisting of the leading* $K-1$ *eigenvectors (ordered in absolute eigenvalue) of* $\boldsymbol{A}^\tau$*.*

3. *For each* $i \in [n]$*, let* $\hat{\boldsymbol{U}}^{K-1}_i$ *denote the transpose of the ith row of*

$\hat{\boldsymbol{U}}^{K-1}$. *Solve the following* $(1+\xi)$*-approximation K-means optimization problem: find some* $\tilde{\boldsymbol{c}}^0_{\mathsf{esc}} = (\tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(1), \cdots, \tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(n))^\top \in [K]^n$, *such that*

$$\begin{aligned}
&\sum_{k=1}^{K} \min_{\boldsymbol{\nu}_k \in \mathbb{R}^{K-1}} \sum_{i:\tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(i)=k} \|\hat{\boldsymbol{U}}_i^{K-1} - \boldsymbol{\nu}_k\|_2^2 \\
&\leqslant (1+\xi) \min_{\boldsymbol{c}\in[K]^n} \sum_{k=1}^{K} \min_{\boldsymbol{\nu}_k \in \mathbb{R}^{K-1}} \sum_{i:\boldsymbol{c}(i)=k} \|\hat{\boldsymbol{U}}_i^{K-1} - \boldsymbol{\nu}_k\|_2^2. \qquad (2.5)
\end{aligned}$$

4. *For each* $k, l \in [K]$, *let*

$$\tilde{P}^0_{kl} = \begin{cases} \dfrac{\sum_{i<j} A^{\tau}_{ij} \mathbb{I}\{\tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(i)=k, \tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(j)=k\}}{\frac{1}{2}\tilde{n}^0_k(\tilde{n}^0_k-1)}, & k = l, \\ \dfrac{\sum_{i,j\in[n]} A^{\tau}_{ij} \mathbb{I}\{\tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(i)=k, \tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(j)=l\}}{\tilde{n}^0_k \tilde{n}^0_l}, & k \neq l, \end{cases}$$

with $\tilde{n}^0_k = \sum_{j\in[n]} \mathbb{I}\{\tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(j) = k\}$. *Let*

$$k^* = \arg\min_{k\in[K]} \tilde{P}^0_{kk}. \qquad (2.6)$$

Then, for each i *with* $\tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(i) = k^*$, *let* $\boldsymbol{c}^0_{\mathsf{esc}}(i) = K$; *for each* i *with* $\tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(i) = K$, *let* $\boldsymbol{c}^0_{\mathsf{esc}}(i) = k^*$; *and let* $\boldsymbol{c}^0_{\mathsf{esc}}(i) = \tilde{\boldsymbol{c}}^0_{\mathsf{esc}}(i)$, *for each remaining node.*

From the simulation results presented in Section 4, it can be seen that the community extraction performance of ESC and INIT is very similar. However, in terms of running time, ESC significantly outperforms INIT, because it only needs to cluster n $(K-1)$-dimensional vectors.

Note that when applying Algorithms 2 and 3, we choose $\tau = 2\bar{d}$, where $\bar{d} = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}/n$ is the average degree of the network. In addition, we use the corrected Bayesian Information Criterion (CBIC) proposed by Hu et al. (2020) for selecting K. In Section S2 of the Supplement, we will discuss in detail how we select τ and K, and present some simulation results for model selection.

3. Theoretical guarantee of algorithms

In this section, we investigate the theoretical properties of the two initialization algorithms INIT and ESC as well as the refinement algorithm RACEn.

3.1 Parameter space and loss function

We consider the case of $K = 2$ as in Zhao et al. (2011) and Wilson et al. (2017), which was regarded as "single extraction". In the case of $K = 2$, for each $\boldsymbol{c} \in [2]^n$ and each $k \in [2]$, let $n_k(\boldsymbol{c}) = \Big|\{i \in [n] : \boldsymbol{c}(i) = k\}\Big|$. Let $\mathcal{C}_0 = \{\boldsymbol{c} : [n] \to [2]^n\}$. We consider the following parameter space for community extraction:

$$\Theta_n(p,q,\beta) = \Big\{(\boldsymbol{P},\boldsymbol{c}) : \boldsymbol{c} \in \mathcal{C}_0,\ n_1(\boldsymbol{c}) \in \Big[\lfloor \beta n \rfloor - 1, \lceil (1-\beta)n \rceil + 1\Big],$$
$$\boldsymbol{P} = \boldsymbol{P}^\top = (P_{kl})_{2\times 2} \in [0,1]^{2\times 2},\ P_{11} = p > q = P_{12} = P_{21} = P_{22}\Big\}, \tag{3.1}$$

where $\beta \in (0, 1/2]$ and $p, q \in (0, 1)$ with $p > q$. Further, we assume that $\beta \gg 1/n$ and $1/n \ll p < 1 - \epsilon_0$ for some small constant $\epsilon_0 \in (0, 1)$.

Remark 1. *The constraint $P_{11} = p > q = P_{12} = P_{21} = P_{22}$ ensures that a background node have relatively few connections to either community and background group, where the nodes with label 2 are background nodes.*

Next, we present the loss function for defining the asymptotic minimax risk. Specifically, for any $\boldsymbol{c} \in \mathcal{C}_0$, the loss function is defined as

$$\ell(\boldsymbol{c}, \hat{\boldsymbol{c}}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{\boldsymbol{c}(i) \neq \hat{\boldsymbol{c}}(i)\}, \text{ for each } \hat{\boldsymbol{c}} \in [2]^n, \tag{3.2}$$

which is the normalized Hamming distance between $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$. This is a misclassification rate of $\hat{\boldsymbol{c}}$ if $\boldsymbol{c}$ is considered as the true label vector.

3.2 Asymptotic minimax risk

For any estimator $\hat{\boldsymbol{c}}$ of $\boldsymbol{c}$, the maximum risk of the SBM in (2.1) based on the parameter space $\Theta_n(p, q, \beta)$ in (3.1) in terms of the loss function ℓ in (3.2) and the estimator $\hat{\boldsymbol{c}}$ is defined as follows:

$$\sup_{(\boldsymbol{P},\boldsymbol{c})\in\Theta_n(p,q,\beta)} \mathbb{E}_{\boldsymbol{P},\boldsymbol{c}}\, \ell(\boldsymbol{c}, \hat{\boldsymbol{c}}). \tag{3.3}$$

Before deriving the asymptotic minimax risk, we need to give some definitions and notation. For any $t \geqslant 0$, define

$$I_t(p, q) = -\log\left\{q^t p^{1-t} + (1-q)^t(1-p)^{1-t}\right\}. \tag{3.4}$$

Then, $I_t(p,q)/(1-t)$ with $t \in (0,1) \cup (1,+\infty)$ is the so-called Rényi-divergence with order t between two Bernoulli distributions Bern(q) and Bern(p), and $I_t(p,q) \equiv I_{1-t}(q,p)$ for any $t \in (0,1)$. Let

$$t^* = t^*(p,q) = \frac{\log\left[\frac{(p-1)\{\log(1-p)-\log(1-q)\}}{p(\log p-\log q)}\right]}{\log\frac{q(1-p)}{p(1-q)}}. \tag{3.5}$$

Due to Lemma 1 provided in the Supplement, we have that t^* is the unique maximum point of $I_t(p,q)$ on $[0,+\infty)$.

Next, we introduce an assumption to prepare for the analysis of the upper and lower bounds of the asymptotic minimax risk.

Assumption 1. *As $n \to \infty$, $-\beta^2 n I_{t^*}(p,q)/\log\beta \to \infty$.*

Remark 2. *Assumption 1 assumes that as $n \to \infty$, β can be any constant in $(0,1/2]$. β can also go to zero, but cannot go to zero too fast. β can go to 0 with certain rate, implies that the number of community nodes can have an order of magnitude difference from the number of background nodes. On the other hand, Assumption 1 assumes that $I_{t^*}(p,q)$ cannot be too small, indicating that there must be a significant gap between the connectivity within the community and the connectivity beyond the community.*

Assume that network $\boldsymbol{A}$ is generated from model (2.1) with parameter $(\boldsymbol{P},\boldsymbol{c}) \in \Theta_n(p,q,\beta)$. Below, we will use a two-step estimation method to get an upper bound of the maximum risk. Specifically, we use a slightly modified version of RACEn, where in the refinement step, we replace $\hat{p}^{0i}$ and $\hat{q}^{0i}$ in RACEn with p and q, respectively. Then, define $\hat{\boldsymbol{c}}_{p,q} = (\hat{\boldsymbol{c}}_{p,q}(1), \cdots, \hat{\boldsymbol{c}}_{p,q}(n))^\top$ with

$$\hat{\boldsymbol{c}}_{p,q}(i) = \begin{cases} 1, & L(\boldsymbol{A}_i; \boldsymbol{c}^{0i}, p) > L(\boldsymbol{A}_i; \boldsymbol{c}^{0i}, q), \\ 2, & \textit{otherwise}, \end{cases} \tag{3.6}$$

where $L(\boldsymbol{A}_i; \boldsymbol{c}^{0i}, x) = \sum_{j \neq i : \boldsymbol{c}^{0i}(j)=1} \{A_{ij} \log x + (1 - A_{ij}) \log(1-x)\}$ for $x \in (0, 1)$ and $\boldsymbol{A}_i$ denotes the i-th row of $\boldsymbol{A}$.

Condition 1. *For a given positive sequence $\{\gamma_n\}$, there exists a constant $C_0 > 0$, such that*

$$\inf_{(\boldsymbol{P},\boldsymbol{c}) \in \Theta_n(p,q,\beta)} \min_{i \in [n]} \mathbb{P}_{P,\boldsymbol{c}} \left\{ \ell\left(\boldsymbol{c}_{-i}, \boldsymbol{c}^0_{-i}\right) \leqslant \gamma_n \right\} \geqslant 1 - n^{-(1+C_0)}, \tag{3.7}$$

where $\boldsymbol{c}_{-i} = (\boldsymbol{c}(1), \cdots, \boldsymbol{c}(i-1), \boldsymbol{c}(i+1), \cdots, \boldsymbol{c}(n))^\top$, $\boldsymbol{c}^0$ is obtained by an initialization algorithm, and $\boldsymbol{c}^0_{-i} = (\boldsymbol{c}^0(1), \cdots, \boldsymbol{c}^0(i-1), \boldsymbol{c}^0(i+1), \cdots, \boldsymbol{c}^0(n))^\top \in [2]^{n-1}$ is the output of the initialization algorithm $\boldsymbol{c}^0$ applied to $\boldsymbol{A}_{-i}$.

Condition 1 requires that in each of the n implementations of the initialization algorithm, the loss is at most γ_n almost surely. This requirement imposes a certain uniform consistency condition on the estimators of the group labels obtained by the initialization algorithm.

Based on the above definition, we can obtain the following proposition.

Proposition 1. *Suppose that as $n \to \infty$, $\beta n I_{t*} \to \infty$ and $\boldsymbol{c}^0$ obtained by the initialization algorithm satisfies Condition 1 with $\gamma_n = o(\beta)$ when $\lim_{n\to\infty} p/q > 1$, and with $\gamma_n = o(-\beta(p-q)/p)$ when $\lim_{n\to\infty} p/q = 1$. If $\lim_{n\to\infty} \beta n I_{t*}(p,q)/\log n \leqslant 1$, then*

$$\limsup_{n\to\infty} \frac{1}{\beta n I_{t*}(p,q)} \log\left\{\sup_{(\boldsymbol{P},\boldsymbol{c})\in\Theta_n(p,q,\beta)} \mathbb{E}_{\boldsymbol{P},\boldsymbol{c}}\ell(\boldsymbol{c},\hat{\boldsymbol{c}}_{p,q})\right\} \leqslant -1. \tag{3.8}$$

If $\lim_{n\to\infty} \beta n I_{t*}(p,q)/\log n > 1$, then $\mathbb{E}\ell(\boldsymbol{c},\hat{\boldsymbol{c}}_{p,q}) \leqslant n^{-(1+C)}$ for some small positive constant C, which means that $\hat{\boldsymbol{c}}_{p,q}$ exactly restored the label $\boldsymbol{c}$ in the expected sense.

Based on Proposition 1, we establish the following theorem, which characterizes the asymptotic behavior of $\hat{\boldsymbol{c}}_{p,q}$ in terms of the resulting maximum risk that it achieves, when we choose Algorithm 2 as our initialization algorithm, i.e, $\boldsymbol{c}^0 = \boldsymbol{c}^0_{\mathsf{init}}$.

Theorem 1. *Assume $\left\{\Theta_n(p,q,\beta)\right\}_{n=1}^{\infty}$ satisfies Assumption 1 and $\lim_{n\to\infty} p/q > 1$. If $\lim_{n\to\infty} \beta n I_{t*}(p,q)/\log n \leqslant 1$, then*

$$\limsup_{n\to\infty} \frac{1}{\beta I_{t*}(p,q)} \log\left\{\sup_{(\boldsymbol{P},\boldsymbol{c})\in\Theta_n(p,q,\beta)} \mathbb{E}_{\boldsymbol{P},\boldsymbol{c}}\ell(\boldsymbol{c},\hat{\boldsymbol{c}}_{p,q})\right\} \leqslant -1. \tag{3.9}$$

If $\lim_{n\to\infty} \beta n I_{t*}(p,q)/\log n > 1$, then $\mathbb{E}\ell(\boldsymbol{c},\hat{\boldsymbol{c}}_{p,q}) \leqslant n^{-(1+C)}$ for some small positive constant C, which means that $\hat{\boldsymbol{c}}_{p,q}$ exactly restored the label $\boldsymbol{c}$ in the expected sense. When $\lim_{n\to\infty} p/q > 1$ is replaced by $\lim_{n\to\infty} p/q = 1$, if an *additional condition* $-(p-q)\beta^2 nI_{t*}(p,q)/(p\log\beta)\to\infty$ *is added, the above conclusion still holds.*

Next, we show that the maximum risk of $\hat{\boldsymbol{c}}_{p,q}$ established in Theorem 1 is the best we can hope to achieve, i.e., it matches the asymptotic minimax lower bound. To do this, we first establish the minimax lower bound in Theorem 2.

Theorem 2. *Assume the parameter space sequence* $\Big\{\Theta_n(p,q,\beta)\Big\}_{n=1}^{\infty}$ *satisfies Assumption 1. Then, when* $p\asymp q$,

$$\liminf_{n\to\infty}\frac{1}{\beta nI_{t*}(p,q)}\log\Big\{\inf_{\hat{\boldsymbol{c}}}\sup_{(\boldsymbol{P},\boldsymbol{c})\in\Theta_n(p,q,\beta)}\mathbb{E}_{\boldsymbol{P},\boldsymbol{c}}\,\ell(\boldsymbol{c},\hat{\boldsymbol{c}})\Big\}\geqslant -1. \tag{3.10}$$

When $p\asymp q$ *is replaced by* $p\gg q$, *if additional conditions*

$$p\log^3\left(\frac{p}{q}\right)<\infty,\quad \lim_{n\to\infty}\frac{\log\frac{\log\frac{p}{q}}{p}}{\log n}<1 \quad \textit{and} \quad \lim_{n\to\infty}\frac{\log\beta np}{\log\log\frac{p}{q}}>3$$

are added, (3.10) *still holds.*

Then, combining Theorems 1 and 2, we immediately obtain the asymptotic minimax risk for community extraction, which is presented in the following corollary.

Corollary 1. *Assume the parameter space sequence* $\Big\{\Theta_n(p,q,\beta)\Big\}_{n=1}^{\infty}$ *satisfies both the conditions of Theorem 1 and Theorem 2. Then,*

$$\inf_{\hat{\boldsymbol{c}}}\sup_{(\boldsymbol{P},\boldsymbol{c})\in\Theta_n(p,q,\beta)}\mathbb{E}_{\boldsymbol{P},\boldsymbol{c}}\,\ell(\boldsymbol{c},\hat{\boldsymbol{c}})=\exp\Big\{-\big(1+o(1)\big)\beta nI_{t*}(p,q)\Big\}. \tag{3.11}$$

Remark 3. *It can be seen that the asymptotic minimax risk rate for community extraction, i.e.* $I_{t*}(p,q)$*, and that for community discovery, i.e.* $2I_{1/2}(p,q)$*, are very different by the fact* $I_{1/2}(p,q) < I_{t*}(p,q) < 2I_{1/2}(p,q)$*. The reasons of this difference are listed as follows:* (1) *when* $K = 2$*,* $P_{11} = p$ *and* $P_{12} = P_{22} = q$*, the most unfavorable scenario for* $\boldsymbol{c}$ *in our parameter space is the case that* $n_1(c) = \beta n$*. However, in Zhang and Zhou (2016),* $P_{11} = P_{22} = p$ *and* $P_{12} = q$*, and the most unfavorable scenario for* $\boldsymbol{c}$ *is the case that* $n_1(\boldsymbol{c}) = n_2(\boldsymbol{c}) = n/2$*, where the node numbers of different groups are balanced.* (2) *We do not have the symmetry property that* $(P_{11}, P_{12}) = (p,q)$ *and* $(P_{21}, P_{22}) = (q,p)$*. These two differences not only lead to the difference of the minimax risk rates, but also make the work of establishing the theoretical results for community extraction more difficult.*

3.3 Theoretical guarantee of RACEn

Below, we ill establish the property that the output $\check{\boldsymbol{c}}$ of Algorithm 1 achieves the asymptotic minimax risk in (3.11).

Theorem 3. *Suppose that* $\beta n(p-q)^4/p \to \infty$*,* $\beta n I_{t*}(p,q) \to \infty$*,* $p \asymp q$ *as* $n \to \infty$*, and the initialization algorithm in Algorithm 1 satisfies Condition 1 with*

$$\gamma_n = o\left\{-\frac{\beta}{\log \beta}(p-q)\right\}. \tag{3.12}$$

Then, there is a sequence $\eta \to 0$, such that

$$\sup_{(\boldsymbol{P},\boldsymbol{c})\in\Theta_n(p,q,\beta)} \mathbb{P}_{\boldsymbol{P},\boldsymbol{c}}\left[\ell(\boldsymbol{c},\check{\boldsymbol{c}}) \geqslant \exp\left\{-(1+\eta)\beta n I_{t^*}(p,q)\right\}\right] \to 0, \tag{3.13}$$

where $\check{\boldsymbol{c}}$ is the output of Algorithm 1. When $p \asymp q$ is replaced by $p \gg q$, if an additional condition

$$\gamma_n = o\left(-\frac{\beta}{\log\beta} q \frac{\log\log\frac{p}{q}}{\log\frac{p}{q}}\right) \tag{3.14}$$

is added, (3.13) *still holds.*

Theorem 3 shows that the community extraction result of Algorithm 1 reaches the asymptotic minimax risk of community extraction we established.

3.4 Consistency property of the initial algorithms

First, we establish the consistency property of Algorithm 2 (INIT).

Theorem 4. *Let $K = 2$. Suppose that as $n \to \infty$, $-\beta^2 n(p-q)^2/(p\log\beta) \to \infty$. Let $\tau = C_1(np+1)$ for some sufficiently large constant $C_1 > 0$. Then, the output of Algorithm 2, i.e. $\boldsymbol{c}^0_{\mathsf{init}}$, satisfies*

$$\inf_{(\boldsymbol{P},\boldsymbol{c})\in\Theta_n(p,q,\beta)} \mathbb{P}_{P,\boldsymbol{c}}\left\{n\ell\left(\boldsymbol{c},\boldsymbol{c}^0_{\mathsf{init}}\right) \leqslant C(1+\xi)\frac{np+1}{\beta n(p-q)^2}\right\} \geqslant 1-n^{-(1+C')},$$

for constants $C, C' > 0$, where ξ comes from the $(1+\xi)$-approximation K-means optimization in step 3 *of Algorithm 2.*

Below, we will establish the consistency property of ESC.

Theorem 5. *Let* $(\boldsymbol{P}, \boldsymbol{c}) \in \Theta_n(p,q,\beta)$ *and assume that as* $n \to \infty$, $-\beta\Delta^2(\lambda_1 - \lambda_2)^2/(p\log\beta) \to \infty$, *where* λ_1 *and* λ_2 *are the first and second largest eigenvalues of* $\boldsymbol{M}' = (P_{\boldsymbol{c}(i)\boldsymbol{c}(j)})_{n\times n}$, *respectively, and*

$$\Delta^2 = \frac{1}{(\tilde{x}+z)^2 + y^2}\left\{\frac{1}{\sqrt{n_1(\boldsymbol{c})}}(\tilde{x}+z) - \frac{1}{\sqrt{n_2(\boldsymbol{c})}}y\right\}^2$$

with

$$\tilde{x} = \frac{n_1(\boldsymbol{c})p - n_2(\boldsymbol{c})q}{2},\ y = \sqrt{n_1(\boldsymbol{c})n_2(\boldsymbol{c})}q \text{ and } z = \sqrt{\tilde{x}^2 + y^2}.$$

Then, there exist two constants $C, C' > 0$, *such that*

$$\mathbb{P}_{P,\boldsymbol{c}}\left\{\ell\left(\boldsymbol{c}, \boldsymbol{c}^0_{\mathsf{esc}}\right) \leqslant C(1+\xi)\frac{p}{\Delta^2(\lambda_1-\lambda_2)^2}\right\} \geqslant 1 - n^{-(1+C')},$$

where $\boldsymbol{c}^0_{\mathsf{esc}}$ *is the output of Algorithm 3, and* ξ *comes from the* $(1+\xi)$*-approximation K-means optimization in step* 3 *of Algorithm 3.*

By simple calculation, we see that $\lim_{n\to\infty} p/q > 2(1-\beta)/\beta$ is a sufficient condition for $\Delta > 0$.

4. Simulation studies

In this section, we compare the performance of the refinement algorithm RACE initialized with the two initialization algorithms INIT and ESC, respectively, with some of their competitors, including a multilayer extraction algorithm based on modularity that was proposed by Wilson et al. (2017) (abbreviated as M-E), a spectral clustering on ratios-of-eigenvectors that was proposed by Jin (2015) (abbreviated as SCORE), a convexified modularity maximization approach for estimating the communities under degree-corrected block models that was proposed by Chen et al. (2018) (abbreviated as CMM), and a two-stage algorithm to deal with the community detection under degree-corrected block models that was proposed by Gao et al. (2018) (abbreviated as Gao). To make Gao comparable to our algorithms, we have made it have the same initialization algorithms, i.e. INIT and ESC, as ours, and the resulting two-stage algorithms are abbreviated as Gaoinit and Gaoesc, respectively. Similarly, RACEinit and RACEesc denote the refinement algorithms based on RACE and initialized with INIT and ESC, respectively. All methods are implemented in software R and run on a single processor with an Intel(R) Xeon(R) E5-2620 CPU of 2.10 GHz.

We consider the simulation setting used in Li et al. (2020) under the framework of stochastic block model, where all simulations are repeated 100 times. Specifically, we generate a network with n nodes containing $K-1$

communities. For any nonnegative constants p_0, q_0 and q_0', let

$$\boldsymbol{P}_0 = \begin{bmatrix} \boldsymbol{P}_{K-1} & q_0\mathbf{1}_{K-1} \\ \boldsymbol{q}_0\mathbf{1}_{K-1}^\top & q_0 \end{bmatrix},$$

where $\boldsymbol{P}_{K-1} = (P_{K-1,kl})_{(K-1)\times(K-1)}$ with $P_{K-1,kk} = p_0$ and $P_{K-1,kl} = q_0'$ for $k \neq l \in [K-1]$ and $\mathbf{1}_{K-1} = (1, \cdots, 1)^\top \in \mathbb{R}^{K-1}$. Let $\boldsymbol{P} = d\boldsymbol{P}_0/\big(n\omega^\top \boldsymbol{P}_0\omega - \omega^\top \mathrm{diag}(\boldsymbol{P}_0)\big)$, where $\mathrm{diag}(\boldsymbol{P}_0)$ is a K-dimensional vector composed of the diagonal elements of the matrix $\boldsymbol{P}_0$, $d \in \mathbb{R}^+$ is the expected average degree of the network and $\boldsymbol{\omega} = (\omega_1, \cdots, \omega_K)^\top \in [0,1]^K$ with $\sum_{k=1}^K \omega_k = 1$ is the proportional vector composed of the proportions of the network nodes belonging to the communities as well as the background nodes. Given the label vector $\boldsymbol{c} = (\boldsymbol{c}(1), \cdots, \boldsymbol{c}(n))^\top$, the edges A_{ij}'s are generated as independent Bernoulli variables with probabilities proportional to $P_{\boldsymbol{c}(i)\boldsymbol{c}(j)}$'s, respectively.

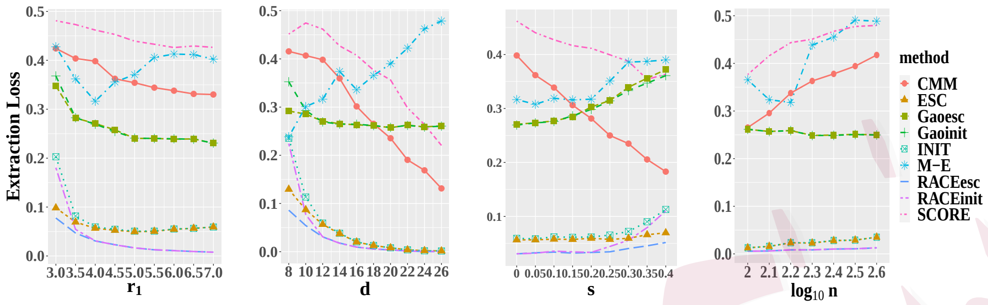
First, we consider the case of only one community, i.e. $K = 2$. Let $n = 100$, $d = 8$ and $\boldsymbol{\omega} = (1-s, 1+s)^\top/2$. Note that $q_0 \equiv q_0'$ when $K = 2$, and hence, we consider the following four settings:

(I) $q_0' = q_0 = 1$, $p_0 = r_1\, q_0$, r_1 varies from 3 to 7 and $s = 0$;

(II) d varies from 8 to 26, $p_0 = 4$, $q_0 = q_0' = 1$ and $s = 0$;

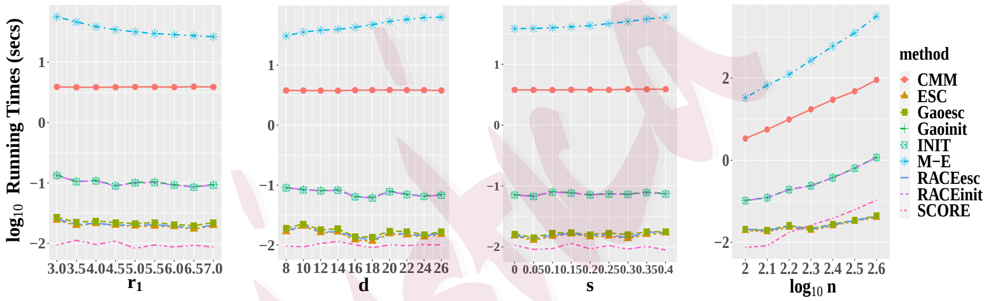
(III) s varies from 0 to 0.4, $p_0 = 4$ and $q_0 = q_0' = 1$;

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0372



1(a) Setting (I) 1(b) Setting (II) 1(c) Setting (III) 1(d) Setting (IV)

Figure 1: The performance of community extraction in case of $K = 2$ for Settings (I)-(IV).



2(a) Setting (I) 2(b) Setting (II) 2(c) Setting (III) 2(d) Setting (IV)

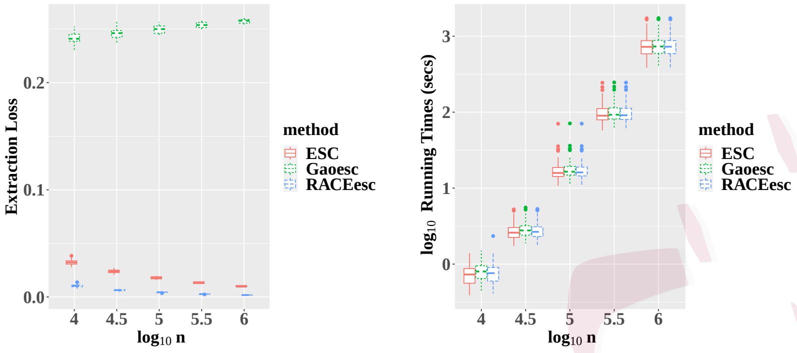
Figure 2: The running times of community extraction in case of $K = 2$ for Settings (I)-(IV).

(IV) $\log_{10} n$ varies from 2 to 2.6, $p_0 = 4$, $q_0 = q_0' = 1$, $d = 12$ and $s = 0$.

In Settings (I)-(IV), we investigate the community extraction performance of the proposed methods and their competitors by varying the values of some involved parameters, respectively. The simulation results are summarized in Figures 1 and 2. From Figure 1, we can see that INIT, ESC, RACEinit and RACEesc outperform M-E, SCORE, CMM, Gaoinit and Gaoesc for all the above settings in terms of the extraction loss defined in (3.2). We notice that for Setting (IV), the performance of M-E deteriorates with the increase of the expected average degree d. According to our experience, this is because M-E tends to extract all the network nodes when d is relatively large. Besides, note that even though Gaoinit and Gaoesc used the same initialization algorithms as RACEinit and RACEesc, respectively, they still did not perform very well because the existence of background nodes was not considered in the refinement step of Gao et al. (2018). Similarly, without considering the presence of background nodes in the network, SCORE and CMM also perform poorly in community extraction, which demonstrates the necessity of developing algorithms for community extraction.

Figure 2 suggests that M-E and CMM are much more time consuming than the other algorithms. For Setting (IV) in Figure 2, the running time of M-E and CMM increases rapidly with the increase of n. Overall, SCORE, ESC, Gaoesc and RACEesc are in the first tier of running speed.

To further demonstrate the advantages of the proposed methods in computational efficiency for dealing with large-scale networks, such as the networks with $n \in [10^4, 10^6]$, below we only compare the following algorithms:



3(a) The extraction loss

3(b) The running time

Figure 3: The performance for large networks in the case of $K = 2$.

ESC, Gaoesc and RACEesc. Specifically, set $K = 2$, $p_0 = 4$, $q_0 = q_0' = 1$, $s = 0$, $d = 5\log_{10} n$ and let $\log_{10} n$ vary from 4 to 6. The obtained results are summarized in Figure 3, which suggest that all these algorithms can deal with large-scale networks with millions of nodes, and RACEesc has much higher community extraction accuracy than ESC and Gaoesc.

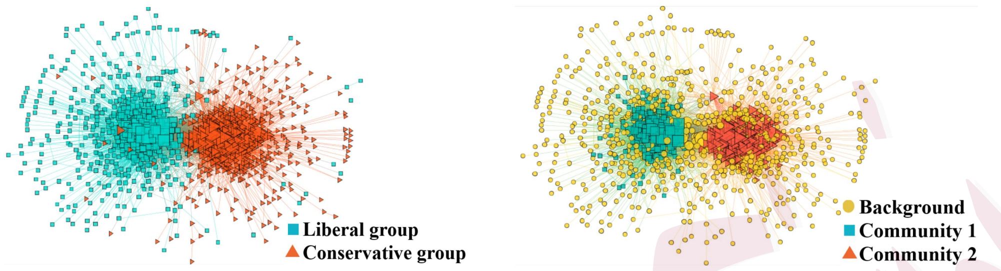
Moreover, we also compare the performance of RACE initialized with the two initialization algorithms INIT and ESC, respectively, with some of their competitors in situation of $K = 3$, which is included in Section S4 of Supplementary Materials.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0372

5. Application

We apply RACEinit to the political blog network, which is commonly studied in the community literatures (Adamic and Glance, 2005; Wang et al., 2020). The nodes of this network are blogs related to US politics and the edges are hyperlinks between blogs. The original network contains $1,490$ nodes. We ignored the directions of the hyperlinks and focused on the largest connected component of the original network as in Karrer and Newman (2011), hence obtained a pre-disposed network of blogs containing $1,222$ nodes and $16,714$ edges. By using the method "corrected Bayesian information criterion" (CBIC) proposed by Hu et al. (2020), we select K as 3.

As shown in Figure 4(1), all blogs in the pre-disposed network were manually labeled as liberal or conservative. In many studies on community discovery of this network, such as Amini et al. (2013) and Wang et al. (2020), researchers often regard the political party labels as the ground truth community labels. In contrast, in this paper, we adopt a new perspective, the perspective of community extraction, to re-explore the community structure of this network. From Figure 4(2), we can see that the blogs labeled as either liberal or conservative can be clearly divided into two groups: core members and non-core members, in which core members have strong internal connections, but non-core members have very weak connections with

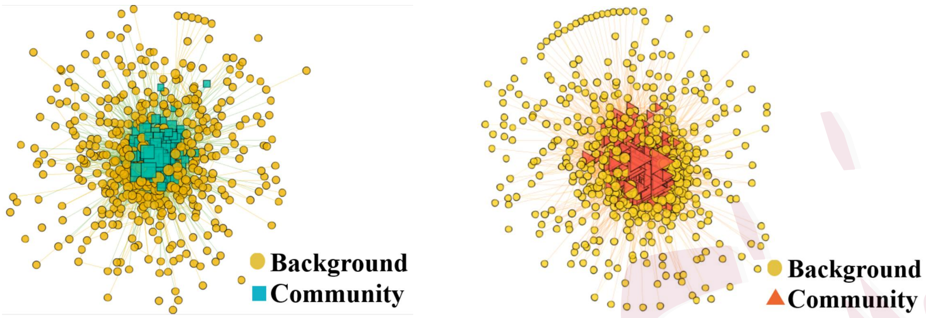


4(a) Liberal group v.s. conservative group 4(b) Apply RACEinit to the whole network with $K = 3$

Figure 4: Visualization of the whole network

both core members and other non-core members.

Indeed, such situation is a suitable example for the community extraction framework studied in this paper. By applying RACEinit to the sub-network composed of the members of each political party with $K = 2$ respectively, we can extract one community from each sub-network, presented in Figure 5, where the community nodes can be viewed as the core members, while the background nodes can be viewed as non-core members. Furthermore, we plot the adjacency matrices of the two sub-networks in Figures 6(1)-(2), where the rows/columns are sorted with respect to the community nodes versus the background nodes. These reordered adjacency matrices can demonstrate significant differences between community nodes



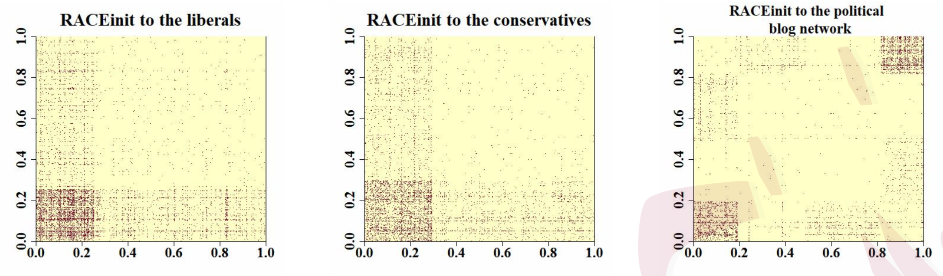
5(a) Apply RACEinit to the liberal sub-network with $K = 2$

5(b) Apply RACEinit to the conservative sub-network with $K = 2$

Figure 5: Visualization of the sub-networks

and background nodes.

The above analysis indicates that there may be a large number of background nodes in the political blog network that should not be ignored, hence we decide to use the proposed community extraction method to analyze the network. Recalling that in the above analysis, we extracted one community from each sub-network, hence here we set $K = 3$, which means that there may be two communities plus additional background nodes. The community extraction results obtained by applying RACEinit to the whole network are presented in Figure 4(2) from the network visualization view and Figure 6(3) from the adjacency matrix view, respectively. In Figure 4(2), Communities 1 and 2 are the extracted communities by RACEinit, which overlap much with the two communities extracted in the sub-networks in Figure 5,



6(a) Apply RACEinit to the liberal sub-network with $K = 2$

6(b) Apply RACEinit to the conservative sub-network with $K = 2$

6(c) Apply RACEinit to the whole network with $K = 3$

Figure 6: The adjacent matrices of the sub-networks and the whole network

i.e. the two core groups extracted from the two sub-networks correspond to the two political parties, respectively. In addition, the background nodes are mainly composed of the non-core members of the two sub-networks. In addition, we plot the adjacency matrix of the whole network in Figure 6(3), where the rows/columns are sorted in the order of community 1, the group of background nodes and community 2. Figure 6(3) suggests that each of the extracted communities has a much stronger connection within itself than with the remaining nodes.

In addition, Table 1 suggests that Communities 1-2 roughly correspond to the core groups of the two subgraphs in Figure 6, and Backgrounds roughly correspond to the corresponding non-core groups.

Table 1: Relationship between Communities 1, 2, backgrounds, and the core groups and the non-core groups obtained from the two sub-networks based on RACEinit.

	Liberal group		Conservative group	
	Core	Non-core	Core	Non-core
Community 1	146	78	0	2
Community 2	0	3	186	49
Background	3	356	3	396

6. Conclusion

In this paper, we proposed some algorithms for community extraction, which are applicable to large-scale networks. We established the asymptotic minimax risk of the SBM for community extraction, based on a specific parameter space with weaker constraints than the parameter space studied in Zhao et al. (2011). Under certain conditions, the proposed algorithm reaches the asymptotic minimax risk, when it is initialized by a low rank approximation algorithm or a spectral clustering algorithm. Then, we demonstrated the advantages of the proposed algorithms via extensive simulation results and a practical application.

Like existing theoretical studies on community extraction (Zhao et al., 2011; Wilson et al., 2017), the theoretical results in this paper is established in the case of $K = 2$. Indeed, establishing the theoretical results for community extraction in the case of $K > 2$ is much more challenging, where nodes in different communities need to be distinguished, in additional to the need to distinguish the community nodes from the background nodes. We leave this challenge as future work.

Acknowledgments

Yuan, Liu and Li's research was partially supported by the National Natural Science Foundation of China grants 12171079, 12101116 and the National Key R&D Program of China grant 2020YFA0714102. Ma's research was partially supported by grants from national sciences foundation and national institute of health.

Supplementary Material

Below we have listed the contents of Supplementary Material. In Section S1, we propose an accelerated refinement algorithm RACE and indicate that the performance of RACE and RACEn is very similar via some simulation results. In Section S2, we demonstrate in detail how we select the tuning parameters τ and K. In Section S3, we explain and compare the assumptions of the main theorems and corollary imposed. In Section S4, we compare the performance of RACE initialized with the two initialization algorithms INIT

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0372

and ESC, respectively, with some of their competitors in situation of $K = 3$. In Section S5, we make some additional discussions. Then, in Section S6, we present the proofs of Theorems 1-5, Proposition 1 and Corollary S1.

References

Adamic, L. A. and N. Glance (2005). The political blogosphere and the 2004 u.s. election: divided they blog. *Proceedings of the 3rd International Workshop on Link Discovery 5*, 36–43.

Amini, A. A., A. Chen, P. J. Bickel, and E. Levina (2013). Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics 41*(4), 2097–2122.

Chen, Y., X. Li, and J. Xu (2018). Convexified modularity maximization for degree-corrected stochastic block models. *The Annals of Statistics 46*(4), 1573–1602.

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

Flake, G. W., S. Lawrence, C. L. Giles, and F. M. Coetzee (2002). Self-organization and identification of web communities. *IEEE Computer 35*, 66–71.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports 486*(3), 75–174.

Fortunato, S. and D. Hric (2016). Community detection in networks: A user guide. *Physics Reports 659*, 1–44.

Gao, C., Z. Ma, A. Zhang, and H. Zhou (2017). Achieving optimal misclassification proportion in stochastic block model. *Journal of Machine Learning Research 18*, 1–45.

Gao, C., Z. Ma, A. Y. Zhang, and H. H. Zhou (2018). Community detection in degree-corrected block models. *The Annals of Statistics 46*(5), 2153–2185.

Goldenberg, A., A. X. Zheng, S. E. Fienberg, and E. M. Airoldi (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning 2*(2), 129–233.

Hartingan, J. and M. K. Wong (1979). Algorithm as136: A k-means clustering algorithm. *Applied statistics 28*, 100–108.

Holland, P. W., K. B. Laskey, and S. Leinhardt (1983). Stochastic block models: First steps. *Social Networks 5*(2), 109–137.

Hu, J., H. Qin, T. Yan, and Y. Zhao (2020). Corrected bayesian information criterion for stochastic block models. *Journal of the American Statistical Association 115*(532), 1771–1783.

Jin, J. (2015). Fast community detection by SCORE. *The Annals of Statistics 43*(1), 57–89.

Karrer, B. and M. E. J. Newman (2011). Stochastic blockmodels and community structure in networks. *Physical Review E 83*, 016107.

Kumar, A., Y. Sabharwal, and S. Sen (2004). A simple linear time $(1+\xi)$-approximation algorithm for k-means clustering in any dimensions. *Proceedings-Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 454–462.

Lei, J. and A. Rinaldo (2015). Consistency of spectral clustering in stochastic block models. *Annals of Statistics 43*(1), 215–237.

Li, T., E. Levina, and J. Zhu (2020). Network cross-validation by edge sampling. *Biometrika 107*(2), 257–276.

Moody, J. and D. R. White (2003). Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review 68*, 103–127.

Newman, M. E. J. and M. Girvan (2004). Finding and evaluating community structure in networks. *Physical Review E 69*(2), 026113.

Shi, J. and J. Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(8), 888–905.

Spirin, V. and L. A. Mirny (2003). Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences 100*(21), 12123–12128.

Wang, F., T. Li, X. Wang, S. Zhu, and C. Ding (2011). Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery 22*(3), 493–521.

Wang, J., J. Zhang, B. Liu, J. Zhu, and J. Guo (2020). Fast network community detection with profile-pseudo likelihood methods. *Journal of the American Statistical Association 118*, 1359–1372.

Wasserman, S. and K. Faust (1994). *Social network analysis: methods and applications*. Structural Analysis in the Social Sciences. Cambridge University Press.

Wei, Y.-C. and C.-K. Cheng (1989). Towards efficient hierarchical designs by ratio cut partitioning. *1989 IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*, 298–301.

Wilson, J. D., J. Palowitch, S. Bhamidi, and A. B. Nobel (2017). Community extraction in multilayer networks with heterogeneous community structure. *Journal of machine learning research 18*, 5458–5506.

Yun, S.-Y. and A. Proutiere (2016). Optimal cluster recovery in the labeled stochastic block model. *Proceedings of the 30th International Conference on Neural Information Processing Systems 29*, 973–981.

Zhang, A. Y. and H. H. Zhou (2016). Minimax rates of community detection in stochastic block models. *The Annals of Statistics 44*(5), 2252–2280.

Zhao, Y. (2017). A survey on theoretical advances of community detection in networks. *Wiley Interdisciplinary Reviews: Computational Statistics 9*(5), e1403.

Zhao, Y., E. Levina, and J. Zhu (2011). Community extraction for social networks. *Proceedings of the National Academy of Sciences 108*(18), 7321–7326.

School of Mathematics and Statistics & KLAS, Northeast Normal University

E-mail: (yuanq214@nenu.edu.cn, liubh100@nenu.edu.cn, lidn040@nenu.edu.cn)

Department of Statistics, Pennsylvania State University

E-mail: (yzm63@psu.edu)