Hindawi
Scientific Programming
Volume 2019, Article ID 9471201, 12 pages
https://doi.org/10.1155/2019/9471201

Research Article

A Bayesian Inference Method Using Monte Carlo Sampling for Estimating the Number of Communities in Bipartite Networks

Guo-Zheng Wang,[1] **Li Xiong**,[1] **and Hu-Chen Liu**[2]

[1] *School of Management, Shanghai University, Shanghai 200444, China*
[2] *College of Economics and Management, China Jiliang University, Hangzhou 310018, China*

Correspondence should be addressed to Hu-Chen Liu; huchenliu@foxmail.com

Received 10 September 2019; Revised 14 November 2019; Accepted 21 November 2019; Published 9 December 2019

Academic Editor: Fabrizio Riguzzi

Community detection is an important analysis task for complex networks, including bipartite networks, which consist of nodes of two types and edges connecting only nodes of different types. Many community detection methods take the number of communities in the networks as a fixed known quantity; however, it is impossible to give such information in advance in real-world networks. In our paper, we propose a projection-free Bayesian inference method to determine the number of pure-type communities in bipartite networks. This paper makes the following contributions: (1) we present the first principle derivation of a practical method, using the degree-corrected bipartite stochastic block model that is able to deal with networks with broad degree distributions, for estimating the number of pure-type communities of bipartite networks; (2) a prior probability distribution is proposed over the partition of a bipartite network; (3) we design a Monte Carlo algorithm incorporated with our proposed method and prior probability distribution. We give a demonstration of our algorithm on synthetic bipartite networks including an easy case with a homogeneous degree distribution and a difficult case with a heterogeneous degree distribution. The results show that the algorithm gives the correct number of communities of synthetic networks in most cases and outperforms the projection method especially in the networks with heterogeneous degree distributions.

1. Introduction

A bipartite network is a network with nodes of two types and edges connecting only nodes of different types. The decomposition of bipartite networks into communities (clusters, modules, or groups), i.e., community detection, plays an important role in revealing the structure of large networked systems, providing new insights into how the network is organized [1–4].

Many methods [5–9] have been developed for community detection in bipartite networks in recent years. A fundamental shortcoming of most community detection methods is that they partition networks into a fixed number of groups. However, this number is usually unknown in real-world networks, and we need to mine such information from the network data. A lot of research [10–13] for making such efforts to determine the number of communities in bipartite networks has been proposed recently. There are three main problems in these methods. One is that they performed estimation through maximizing the modularity proposed in [10, 14] that is proved to be NP-hard [15, 16]; the second is that they gave the number of communities of mixed-type, which is nearly always substantially less efficient [6]; the third is that the projection method [15] performed poorly due to information loss. The heuristic methods proposed in [17, 18] for community detection in bipartite networks does not need the number of communities to be given a priori.

In this paper, we propose a projection-free Bayesian inference method for determining the number of pure-type communities in a bipartite network. Our method builds mainly on the work as below: (i) the degree-corrected bipartite stochastic block model, proposed by Larremore et al. [6], is used to find the community structure of empirical networks with broad degree distributions; and (ii) the prior probability distribution over divisions of a network into groups and a new prior probability distribution based on a queueing-type process, both proposed by Riolo et al. [4], are used for calculating the number of communities in a unipartite network.

In Section 2, first, we present the first-principle derivation of a practical method, using the degree-corrected bipartite stochastic block model that is able to deal with networks with broad degree distributions, for estimating the number of pure-type communities of bipartite networks. Second, we propose a prior probability distribution over the partition of a bipartite network, with the community-type parameter ensuring that each community is pure type. In Section 3, we design a Monte Carlo algorithm incorporated with our proposed method and prior probability distribution. In the following section, we demonstrate our method on synthetic bipartite networks including an easy case with homogeneous degree distributions and a difficult case with a heterogeneous degree distribution. The results show that the proposed algorithm can determine the correct number of communities and perform better than our projection method in every case.

2. Methods

2.1. Degree-Corrected Bipartite Stochastic Block Model. The stochastic block model is a generative model used to produce networks containing blocks, groups, or communities. This model is very important in network science and is used for recovering the community structure in network data [2, 3]. The classic stochastic block model can be described as follows: divide the number of vertices N into K disjoint communities; any two vertices i and j are connected by an edge with probability $\omega_{g_i g_j}$, which is an entry of a symmetric $K \times K$ matrix, and g_i is the community of vertex i. However, the block model described above finds the community structure merely due to the degree sequence and fails to detect the known communities in a real-world network that has heterogeneous degree distributions [19]. Karrer and Newman [20] extended the classic stochastic block model including heterogeneity in the degrees of vertices and proposed the degree-corrected stochastic block model, which is proved to overcome the problems of the classic block model.

Most stochastic block model community detection methods can be naturally applied to bipartite networks [20, 21]. Unfortunately, the stochastic block model often overfits bipartite data by mixing nodes of different types within communities and it is nearly always substantially less efficient [6]. Built on the work of Karrer and Newman [20], Larremore et al. [6] proposed the degree-corrected bipartite stochastic block model, which is employed in our calculations. In the degree-corrected bipartite stochastic block model, a bipartite network G is given with an $N_a \times N_b$ bipartite asymmetric adjacency matrix B, where N_a is the number of nodes of type-a and N_b is the number of nodes of type-b. Let $A = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}$ be the $N \times N$ symmetric adjacency matrix of the network G with $N = N_a + N_b$. The type-a nodes are divided into some number k_a of communities, labeled $1, \ldots, k_a$, and the N_b nodes of type-b are divided into k_b communities, labeled $k_a + 1, \ldots, k_a + k_b$. We express the matrix of community interrelationships as a $k \times k$ matrix, where $k = k_a + k_b$. Let g_i again encode the community node i belongs to. Let t_i be the type of vertex i and T_r be the type of community r, imposing the constraint

$$t_i = T_{g_i}, \tag{1}$$

which indicates that node types and community types must match and ensure that communities will be pure type. We write

$$k_a = \sum_{r=1}^{k} \delta_{T_r,\text{type-}a}, \tag{2}$$

$$k_b = \sum_{r=1}^{k} \delta_{T_r,\text{type-}b}, \tag{3}$$

where δ is the Kronecker delta. Let θ_i control the expected degree of node i and ω_{rs} be the $k \times k$ symmetric matrix of parameters to control the number of edges between communities r and s. Following [4], the normalization of θ_i can be fixed by imposing the constraint

$$\frac{1}{n} \sum_i \theta_i \delta_{g_i,r} = 1, \tag{4}$$

where $n_r = \sum_i \delta_{g_i,r}$ is the number of nodes in community r. Following [22], we let the numbers of edges between nodes i and j follow a Poisson distribution with mean $\theta_i \theta_j \omega_{g_i g_j}$. Enforcing the bipartite constraint of equation (1) produces a restriction on ω:

$$\omega_{rs} = 0, \quad \text{when } T_r = T_S. \tag{5}$$

Given parameters g, k, θ, ω, and T for the specification of the mode, the probability of observing a bipartite network G with adjacency matrix A can be written as

$$P(A \mid g, k, \theta, \omega, T) = \prod_{\substack{i<j \\ t_i \neq t_j}} \frac{\left(\theta_i \theta_j \omega_{g_i g_j}\right)^{A_{ij}}}{A_{ij}!} \exp\left(-\theta_i \theta_j \omega_{g_i g_j}\right). \tag{6}$$

Allowing for the constraint of equation (4), the probability $P(A \mid g, k, \theta, \omega, T)$ can be simplified to the more convenient form of

$$P(A \mid g, k, \theta, \omega, T) = \prod_i \theta_i^{d_i} \times \prod_{\substack{r<s \\ T_r \neq T_s}} \omega_{rs}^{m_{rs}} \exp\left(-n_r n_s \omega_{rs}\right), \tag{7}$$

where d_i is the observed degree of vertexes i and $m_{rs} = \sum_{\substack{i,j \\ t_i \neq t_j}} A_{ij} \delta_{g_i,r} \delta_{g_j,s}$ is the number of edges between communities r and s. We have neglected an overall multiplicative constant in (7) since it cancels out in later calculations. Note that a similar probability given by Larremore et al. [6] has been modified in equation (7) as follows:

(i) The number k of communities, the objective we will estimate, is incorporated as an unknown quantity

(ii) The exponential expression is $-n_r n_s \omega_{rs}$ rather than $-\omega_{rs}$, with the normalization of θ_i under a different constraint condition

Then, we integrate out the irrelevant parameters θ and ω. We assume maximum-entropy (i.e., least informative) prior probability distributions on the parameters θ and ω. For θ, this means a uniform prior probability distribution over the regular simplex of values specified by equation (4). Then, we let the expected value of the edge probability ω_{rs} be equal to the observed average edge probability in the network as a whole: $p = 2m/N^2$, where m is the total number of edges in the bipartite network. Then, the maximum-entropy prior probability distribution is an exponential distribution $P(\omega) = (1/p)e^{-\omega/p}$. We assume the priors to be independent (conditioned on g, k, and T) so that $P(\omega, \theta \mid k, g, T) = P(\omega \mid k, T)P(\theta \mid k, g, T)$ and

$$P(A \mid g, k, T) = \iint P(A \mid g, k, \theta, \omega, T)P(\theta \mid g, k, T)P(\omega \mid k, T)\mathrm{d}\theta\, \mathrm{d}\omega. \tag{8}$$

With these choices of priors, integration is performed on equation (8). Then, we have

$$P(A \mid g, k, T) = \prod_r n_r^{\kappa_r} \frac{(n_r - 1)!}{(n_r + \kappa_r - 1)!} \times \prod_{\substack{r<s \\ T_r \neq T_s}} \frac{m_{rs}!}{(pn_r n_s + 1)^{m_{rs}+1}}, \tag{9}$$

where $\kappa_r = \sum_i d_i \delta_{g_i, r}$ and an overall multiplying constant has been discarded.

2.2. Prior on Community Partitions

2.2. Prior on Community Partitions. Our goal is to estimate the correct values of k_a and k_b for a given bipartite network using this model as the basis for a Bayesian model selection procedure. We have

$$P(g, k, T \mid A) = \frac{P(g, k, T)P(A \mid g, k, T)}{P(A)}, \tag{10}$$

where $P(A \mid g, k, T)$ is given by equation (6), and the probability $P(A)$, which in the denominator of equation (10) has no effect on our results, is unknown but cancels out in later calculations. In this paper, our primary focus is to get the posterior distribution on k through summing over g; then, we choose a value for k and calculate k_a and k_b using equations (2) and (3). Now, we start to choose the prior $P(g, k, T)$, which is often the most important and difficult task of the calculation in the case of Bayesian methods.

2.2.1. Prior $P(g \mid k)$ on Community Partitions. If we know the number of communities k in advanced, let us choose the prior $P(g \mid k)$ *on* community partitions of one type of node. We first employ the most commonly used approach, which is described as follows. The prior on the community partition probabilities γ is uniform under the constraint $\sum_r \gamma_r = 1$, where $\gamma_r \in [0, 1]$, with which nodes are assigned to communities independently at random. We can get a particular community partition with the probability

$$P(g \mid \gamma, k) = \prod_{i=1}^{N} \gamma_{g_i} = \prod_{r=1}^{k} \gamma_r^{n_r}. \tag{11}$$

The values γ_r fall on a regular $(k-1)$-dimensional simplex with volume $(k-1)!$, so its probability density is $P(\gamma \mid k) = (k-1)!$. We integrate equation (11) over the simplex and get the following equation [4, 23, 24]:

$$P(g \mid k) = \int P(g \mid \gamma, k)P(\gamma \mid k)\mathrm{d}\gamma = \frac{(k-1)!}{(N+k-1)!} \prod_r n_r!. \tag{12}$$

Since the process above generates a uniform distribution over possible community sizes, we then introduce an alternative and simpler way (used by Riolo et al. [4]) to derive the prior $P(g \mid k)$. We have $\binom{N+k-1}{k-1}$ possible ways to choose k communities with $\sum_{r=1}^{k} n_r = N$ and $N!/\prod_r n_r!$ possible ways to place the nodes in the k communities; thus, any partition g of nodes to communities is given with the probability

$$P(g \mid k) = \frac{1}{\binom{N-1}{k-1} N!/\left(\prod_r n_r!\right)} = \frac{(k-1)!}{(N+k-1)!} \prod\nolimits_r n_r!, \tag{13}$$

the same to equation (12) without the need for parameters γ_r.

However, these two methods may generate partitions g with empty communities. As in [4], we have the binomial coefficient $\binom{N-1}{k-1}$ possible choices of k communities with nonempty ones and

$$P(g \mid K) = \frac{1}{\binom{N-1}{k-1} N!/\left(\prod_r n_r!\right).} \tag{14}$$

Then, we allow for two different types of partitions and get

$$\begin{aligned} P(g \mid k, T) &= \frac{1}{\binom{N_a-1}{k_a-1} N_a!/\left(\prod_{T_r=\text{type}-a} n_r!\right)} \\ &\quad \cdot \frac{1}{\binom{N_b-1}{k_b-1} N_a!/\left(\prod_{T_r=\text{type}-b} n_r!\right)} \\ &= \frac{1}{\binom{N_a-1}{k_a-1}\binom{N_b-1}{k_b-1} N_a! N_b!/\left(\prod_r n_r!\right)}. \end{aligned} \tag{15}$$

2.2.2. Choice of the Number of Communities. Some previous work has been done for the choices of prior $P(k)$ over the number of communities itself, such as letting $P(k)$ equal $1/N$ [23, 24] or $1/k!$ [21]. We again follow [4] and take a different approach, in which community partitions g and the number of communities k can be generated synchronously. We use a queueing-type mechanism for processing community partitions g. For one type, such as type-a, we order the N_a nodes uniformly at random and the first node is placed in community 1. Then, we place each following node either (a) with probability $1-q$ in the same community as the previous node or (b) with probability q in the next community; for another type, we repeat the process above. This process ensures that all communities generated are not empty.

For type-a, there are $N_a!$ ($N_b!$ for type-b) possible ways to order the nodes, so the probability of each one occurring is the same as $1/N_a!$ ($1/N_b!$ for type-b). If $k = k_a + k_b$ communities are generated finally, we must create $k-2$ new communities ($k_a - 1$ new type-a ones and $k_b - 1$ new type-b ones). Because for one type each node except the first starts a new community with equal probability q, k communities with sizes $n_1, \ldots, n_k$ are generated with the probability:

$$(1-q)^{n_1-1}q(1-q)^{n_2-1}q\cdots(1-q)^{n_{k_a}-1}(1-q)^{n_{k_a+1}-1}q \cdot (1-q)^{n_{k_a+2}-1}q\cdots(1-q)^{n_{k_a+k_b}-1} = q^{k-2}(1-q)^{N-k}. \tag{16}$$

For each community of the same partition g, the nodes can be rearranged in $\prod_r n_r!$ ways. Thus, any given partition is generated in the process with the probability:

$$P(g,k,T) = \frac{1}{N_a!N_b!}q^{k-2}(1-q)^{n-k}\prod_{r=1}^{k} n_r!. \tag{17}$$

Given that $P(g,k,T) = P(k,T)P(g \mid k,T)$,

$$P(k,T) = \frac{P(g,k,T)}{P(g \mid k)} = \binom{N_a - 1}{k_a - 1}\binom{N_b - 1}{k_b - 1}q^{k-2}(1-q)^{N-k}. \tag{18}$$

We let $q = \mu/(N-1)$, where the expected number of new communities created is μ. Then,

$$P(g,k,T) = \frac{(1-q)^N}{q^2 N_a! N_b!}\frac{\mu^k}{(N-\mu-1)^k}\prod_{r=1}^{k} n_r!. \tag{19}$$

In equation (19), $((1-q)^N)/(q^2 N_a! N_b!)$ has no effect on our result and cancels out in later calculations.

As in [4], we let $\mu = 1$ and neglecting constants

$$P(g,k,T) = (N-2)^{-k}\prod_{r=1}^{k} n_r!. \tag{20}$$

Now, equation (10) can be written as

$$P(g,k,T \mid A) = (N-2)^{-k}\prod_r n_r^{\kappa_r} n_r! \frac{(n_r-1)!}{(n_r+\kappa_r-1)!} \times \prod_{\substack{r<s \\ T_r \neq T_s}} \frac{m_{rs}!}{(pn_r n_s+1)^{m_{rs}+1}}, \tag{21}$$

and here we allow for equations (9) and (20).

Unfortunately, it is hard to sum over g since the sum has k^N terms [4]. Instead, we approximate the distribution over k (k_a and k_b according to different types) by Markov chain Monte Carlo sampling.

3. Monte Carlo Algorithm for Bipartite Networks

3.1. Our Algorithm. We design a Monte Carlo algorithm incorporated with the bipartite block model and prior probability distribution discussed above to apply the bipartite networks. We call our algorithm the bipartite network Monte Carlo algorithm (BMCA), and it is built on the unipartite network analysis of Riolo et al. [4]. Our algorithm fulfills the requirements of ergodicity and detailed balance [25].

There are two types of steps used by BMCA:

> Type 1: moving one node from its current community to a different existing community. There are again two types of processes in this type of rearrangement. In the processes of the first type, BMCA decreases the number of communities (k_a or k_b) of the same type as the community whose last node moved, thereby decreasing the value of k by one. In the processes of the second type, the community the node moved from contains more than one node and the move here does not change the number of communities.
>
> Type 2: moving one node to a community newly created. The number of communities (k_a or k_b) of the same type as the community the node moved from and the value of k increase by one.

The two types of steps described above make BMCA meet the requirement of ergodicity.

Detailed balance requires that the rate $R(g,k,T \longrightarrow g',k',T')$ goes from a state (g,k,T) to another state (g',k',T') and the opposite meet

$$\frac{R(g,k,T \longrightarrow g',k',T')}{R(g',k',T' \longrightarrow g,k,T)} = \frac{P(g',k',T' \mid A)}{P(g,k,T \mid A)} = \frac{P(g',k',T')}{P(g,k,T)} \times \frac{P(A \mid g',k',T')}{P(A \mid g,k,T)}, \tag{22}$$

where we allow for equation (10). From (20), we have

$$\frac{P(g',k',T')}{P(g,k,T)} = (N-2)^{k-k'}\frac{\prod_{r=1}^{k} n_r'!}{\prod_{r=1}^{k} n_r!}. \tag{23}$$

We consider $R(g,k,T \longrightarrow g',k',T')$ as $\pi(g,k,T \longrightarrow g',k',T')\alpha(g,k,T \longrightarrow g',k',T')$, where the previous part of the product represents the probability of proposing a move and the latter represents the probability chosen to satisfy the detailed balance condition for accepting the move. Then,

$$\frac{R(g,k,T \longrightarrow g',k',T')}{R(g',k',T' \longrightarrow g,k,T)} = \frac{\pi(g,k,T \longrightarrow g',k',T')}{\pi(g',k',T' \longrightarrow g,k,T)} \times \frac{\alpha(g,k,T \longrightarrow g',k',T')}{\alpha(g',k',T' \longrightarrow g,k,T)}. \tag{24}$$

BMCA is described as follows:

Input: the bipartite adjacency matrix B and the node-type vector $\mathbf{t}_i$.

Initial communities partition: using the process described in Section 2.2.2.

Monte Carlo Sampling:

(1) (a) In each step of BMCA, we carry out a rearrangement of type 1 with probability $1 - 1/(N-1)$. If $k = 2$ (i.e., $k_a = 1$ and $k_b = 1$), we do nothing. Otherwise, when $k > 2$, first, we randomly select a community label r in the range $1, \ldots, k$. If the number of communities of type T_r is more than one, then we randomly select a community of type T_r labels s; otherwise, we turn to communities of another type, from which we randomly reselect a pair of communities, respectively, labels r and s. Then, we randomly select one node from community r and move it to community s. The number $k = k_a + k_b$ of total communities remains constant.

(b) In the process, if community r becomes empty as a result that its last node is removed, the number of communities k decreases by one. In practice, for the type-a communities labels $1, \ldots, k_a$, we can efficiently change the community k_a to have label r and then change the community k to have label k_a; specially, if $r = k_a$, we only perform the latter relabeling. In such a process, the number k_a of communities of type-a decreases by 1 and the number k_b of communities of type-b remains constant. For type-b community labels $k_a + 1, \ldots, k$, we can efficiently change the community k to have label r; specifically, if $r = k$, no relabeling is necessary. In such a process, the number k_b of communities of type-b decreases by 1 and the number k_a of communities of type-a remains constant. The number $k_a = k_a + k_b$ of total communities decreases by 1.

(2) Otherwise, we carry out a rearrangement of type 2 with probability $1/(N-1)$. We randomly select a community label r in the range $1, \ldots, k$. If there is only one node in community r, we do nothing. Otherwise, if $T_r = \text{type} - a$, we change community label $k_a + 1$ to $k + 1$ and create a new empty community $k_a + 1$. Then, we randomly select a node from community r and move it to the newly created community $k_a + 1$. During this process, the number k_a of communities of type-a increases by 1 and the number k_b of communities of type-b remains constant. If $T_r = \text{type} - b$, we simply create a new empty community $k + 1$ and no relabeling is necessary. Then, we randomly select a node from community r and move it to the newly created community $k + 1$; during this process, the number k_b of communities of type-b increases by 1 and the number k_a of communities of type-a remains constant. The number $k = k_a + k_b$ of total communities increases by 1.

(3) We accept the rearrangement proposed above with acceptance probability [4]:

$$\alpha(g,k,T \longrightarrow g',k',T') = \min\left[1, \frac{P(A \mid g',k',T')}{P(A \mid g,k,T)}\right]. \tag{25}$$

(4) Repeat steps 1–3.

Output: the posterior probabilities $P(k_a \mid A)$ and $P(k_b \mid A)$.

Rearrangements of any type are performed, and we always have the following equation (see Table 1):

$$\frac{\pi(g,k,T \longrightarrow g',k',T')}{\pi(g',k',T' \longrightarrow g,k,T)} = \frac{P(g',k',T')}{P(g,k,T)}. \tag{26}$$

Taking into consideration equations (22) and (24), the detailed balance condition can be written as

$$\frac{\alpha(g,k,T \longrightarrow g',k',T')}{\alpha(g',k',T' \longrightarrow g,k,T)} = \frac{P(A \mid g',k',T')}{P(A \mid g,k,T)}. \tag{27}$$

Therefore, our algorithm satisfies the detailed balance with acceptance probability of equation (25) and will sample correctly from the distribution $P(g,k,T \mid A)$.

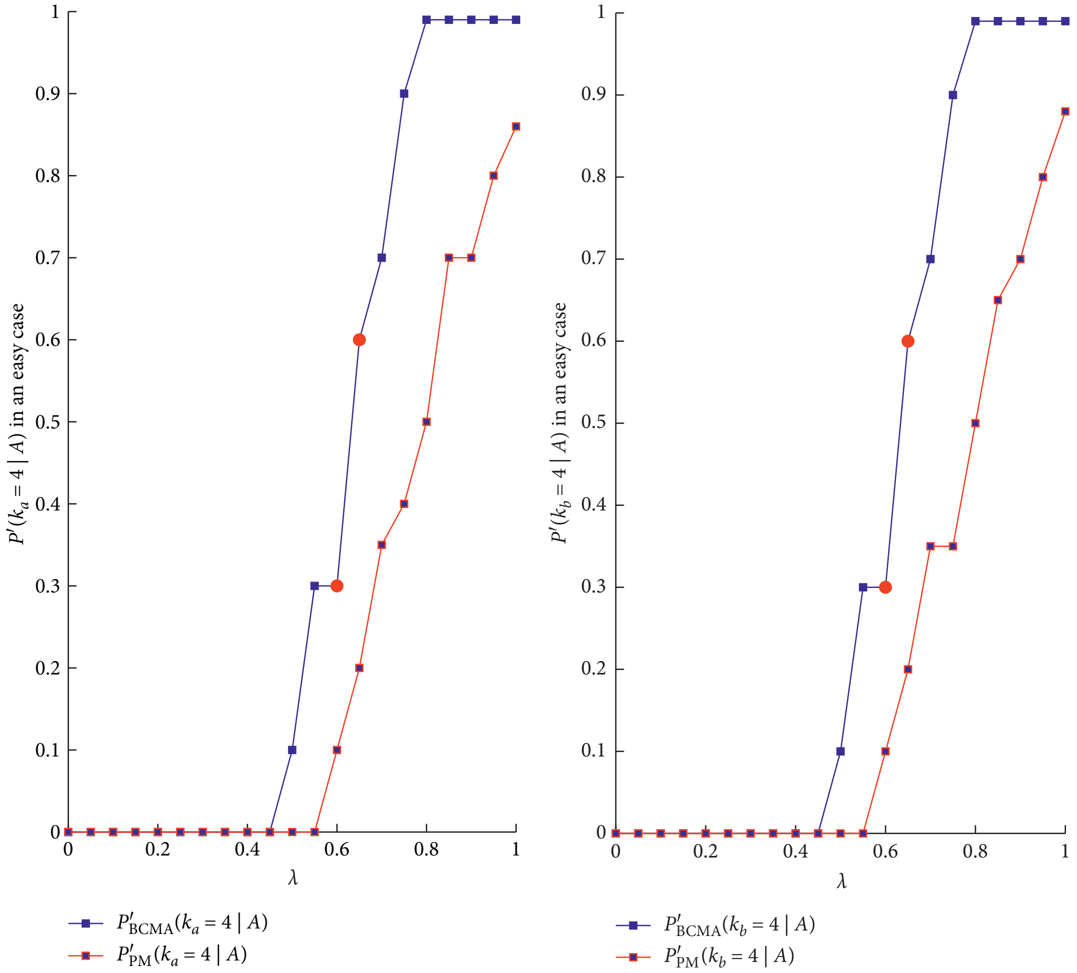
3.2. Output of BMCA. In our implementation, a given number of steps t_i per node is performed in a Monte Carlo run on a bipartite network, and then we write approximate posterior probabilities:

$$P(k,T \mid A) = \frac{\text{the times } k \text{ showed out in a Monte Carlo run}}{S}. \tag{28}$$

For type-a, we can get approximately

Table 1: Derivation of the relation of $(\pi(g,k,T \longrightarrow g',k',T'))/(\pi(g',k',T' \longrightarrow g,k,T))$ and $(P(g',k',T'))/P(g,k,T)$.

Type of rearrangement	States	Probability of performing the move	Probability of selecting two distinct communities of the same type	Probability of selecting a specific node from or to distinct communities of the same type	Total probability π	$(\pi(g,k,T \longrightarrow g',k',T'))/(\pi(g',k',T' \longrightarrow g,k,T))$	$(P(g',k',T'))/(P(g,k,T))$ (using equation (23))
Type 1 with $k' = k$	$g,k,T \longrightarrow g',k',T'$	$1-1/(N-1)$	$1/(k_a(k_a-1)+k_b(k_b-1))$	$1/n_r$	$(1-(1/(N-1))) \times (1/(k_a(k_a-1)+k_b(k_b-1))) \times 1/n_r$	n'_s/n_r	n'_s/n_r
	$g',k',T' \longrightarrow g,k,T$	$1-1/(N-1)$	$1/(k_a(k_a-1)+k_b(k_b-1))$	$1/n'_s$	$(1-(1/(N-1))) \times 1/(k_a(k_a-1)+k_b(k_b-1)) \times 1/n'_s$		
Type 1 with $k' = k-1$	$g,k,T \longrightarrow g',k',T'$	$1-1/(N-1)$	$1/(k_a(k_a-1)+k_b(k_b-1))$	1 (now $n_r = 1$)	$(1-(1/(N-1))) \times 1/(k_a(k_a-1)+k_b(k_b-1))$	$(N-2)n'_s$	$(N-2)n'_s$
	$g',k',T' \longrightarrow g,k,T$	$1/(N-1)$	$1/(k_a(k_a-1)+k_b(k_b-1))$	$1/n'_s$	$(1/(N-1)) \times 1/(k_a(k_a-1)+k_b(k_b-1)) \times 1/n'_s$		
Type 2 with $k' = k+1$	$g,k,T \longrightarrow g',k',T'$	$1/(N-1)$	$1/((k_a+1)k_a+(k_b+1)k_b)$	$1/n_r$	$(1/(N-1)) \times (1/((k_a+1)k_a+(k_b+1)k_b)) \times 1/n_r$	$1/((N-2)n_r)$	$1/((N-2)n_r)$
	$g',k',T' \longrightarrow g,k,T$	$1-1/(N-1)$	$1/((k_a+1)k_a+(k_b+1)k_b)$	1 (now $n'_s = 1$)	$(1-(1/(N-1))) \times (1/(k_a+1)k_a+(k_b+1)k_b)$		



(a)

(b)

Figure 1: Test of BMCA against the projection method (PM) on synthetic networks generated using the bipartite stochastic block model in an easy case. Posterior probabilities $P'(k_a = 4 \mid A)$ (a) and $P'(k_b = 4 \mid A)$ (b) are functions of the mixing parameter λ.

$$P(k_a \mid A) = \frac{\text{the times } k_a \text{ showed out in a Monte Carlo run}}{S}, \tag{29}$$

and for type-b,

$$P(k_b \mid A) = \frac{\text{the times } k_b \text{ showed out in a Monte Carlo run}}{S}. \tag{30}$$

The most likely number of type-a communities in a bipartite network is k_a with the biggest value $P(k_a \mid A)$, and the number of type-b communities k_b can be given in the same way.

In order to avoid any bias in the results and improve the correctness of BMCA, instead of just using $P(k_a \mid A)$, we performed a given number M of Monte Carlo runs for a bipartite network. Thus, we obtained the average value of each $P_i(k_a \mid A)$ as the final posterior probabilities:

$$P'(k_a \mid A) = \frac{\sum_{i=1}^{M} P_i(k_a \mid A)}{M}, \tag{31}$$

for type-a and

$$P'(k_b \mid A) = \frac{\sum_{i=1}^{M} P_i(k_b \mid A)}{M}, \tag{32}$$

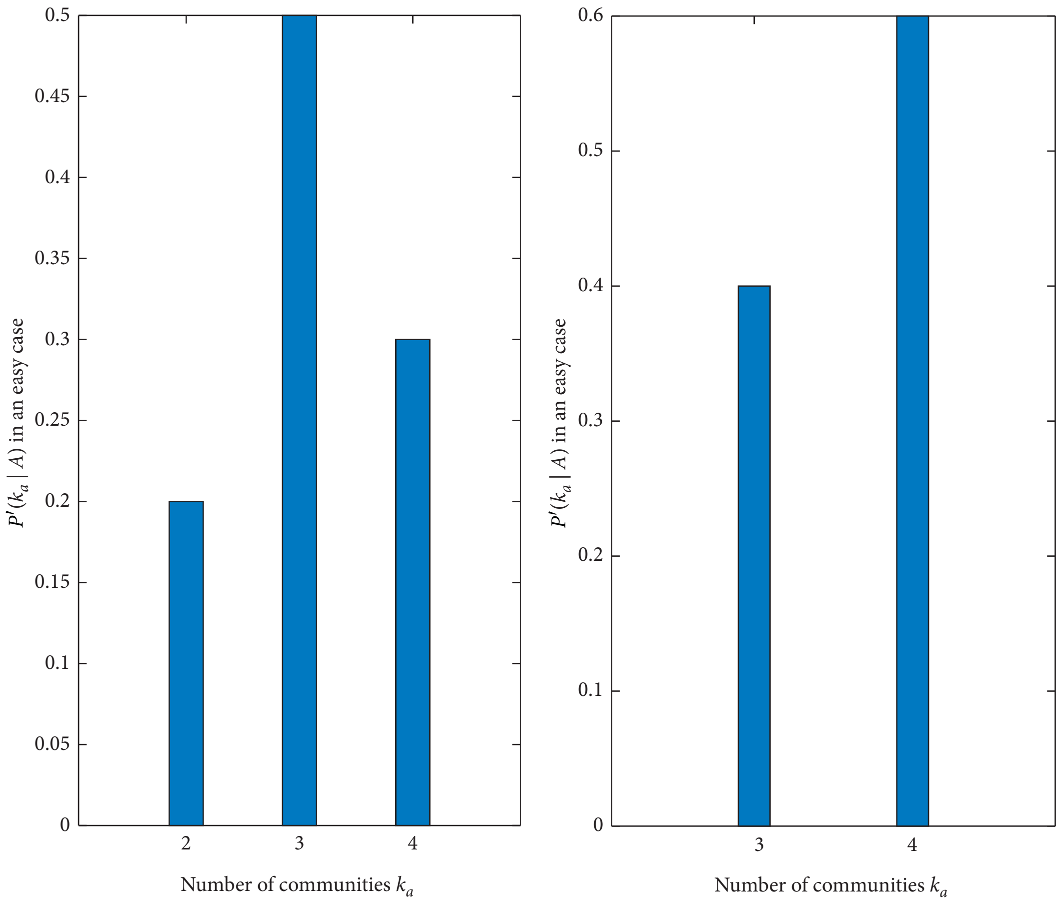
for type-b.

We take $O(N + N_a * N_b)$ to calculate the acceptance probability of each node move, so the time complexity of our algorithm is $O(N_a^2 N_b + N_a N_b^2)$.

4. Example Application

Here, we demonstrate our algorithm on synthetic bipartite networks generated by the bipartite stochastic block model [6] and find that it works well in most cases.

There is often some noise in empirically observed networks because of both errors in the measurements and missing data [26]. Therefore, we employ a mixing model to generate noisy synthetic networks for testing the robustness of our algorithm. In this model, we specify g, and the expected value of an edge is given by



(a)

(b)

Figure 2: Posterior probabilities $P'(k_a \mid A)$ of the number of communities calculated for the synthetic networks generated using the bipartite stochastic block model in an easy case. The synthetic networks were generated with $\lambda = 0.6$ (a) and $\lambda = 0.65$ (b) in the easy case.

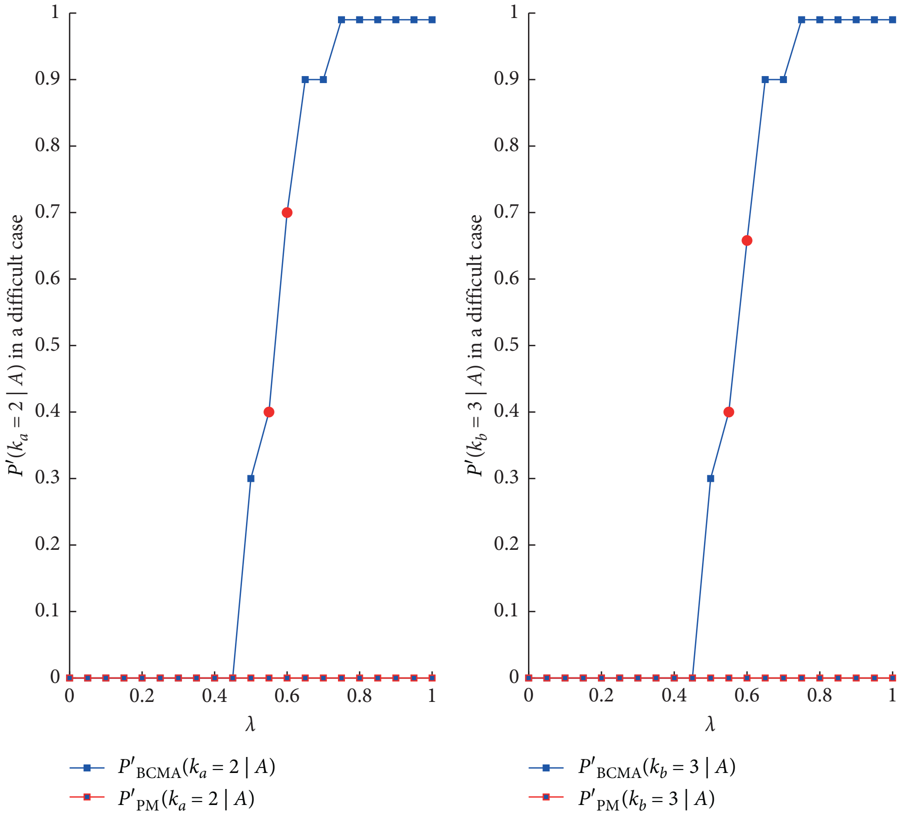
$$\omega = \lambda\omega^{\text{planted}} + (1 - \lambda)\omega^{\text{random}}, \tag{33}$$

where the parameters ω^{planted} and ω^{random} are used to generate a pure planted community structure and no community structure, respectively; and the mixing parameter $\lambda \in [0, 1]$ is used to control various levels of uniformly random noise. Following Larremore et al. [6], we let $\omega_{rs}^{\text{planted}} = m_{rs}$ and $\omega_{rs}^{\text{random}} = \kappa_r\kappa_s/m$, where $\theta_i = d_i/\kappa_{g_i}$. We employ this model to create synthetic networks of an easy case with a homogeneous degree distribution and a difficult case with a heterogeneous degree distribution.

We performed $M = 10$ Monte Carlo runs for each network of $S = 10000$ steps per node and found that BMCA can determine the correct number of communities for synthetic networks and outperform the projection method in every case.

4.1. An Easy Case. In the easy case, we use the model above to create the synthetic networks including four type-a communities and four type-b communities, i.e., $k_a = k_b = 4$, and each node has the same degree (i.e., the network with planted community structure has a homogeneous degree distribution). We let the number of each type of node equal to 1000, and all communities are equally sized as 250. Then, we let $m_{1,5} = m_{2,6} = m_{3,7} = m_{4,8} = 2500$ (i.e., the total number of edges is 10000), and let the block structure matrix ω^{planted} be defined with $\omega_{1,5}^{\text{planted}} = \omega_{2,6}^{\text{planted}} = \omega_{3,7}^{\text{planted}} = \omega_{4,8}^{\text{planted}} = 2500$ (the symmetric entry has the same value). Moreover, the random structure matrix ω^{random} can be defined with $\omega_{i,j}^{\text{random}} = 625$ ($i = 1, 2, 3, 4$; $j = 5, 6, 7, 8$). Finally, with these specifications, we create networks of the easy case to test our algorithm (the code to create these networks can be downloaded from [27]).

As the mixing parameter λ increases, i.e., the level of noise is decreased, BMCA begins to estimate the correct number of communities for the network in an easy case when $\lambda = 0.5$; in addition, the fraction of correct communities number of the network calculated by BMCA increases as a whole (blue line in Figure 1). Then, we use our method to derive the synthetic mixing networks generated with $\lambda = 0.6$ and $\lambda = 0.65$, indicated by red circles in Figure 1(a) and Figure 1(b) and showing the posterior probabilities of the number of communities in the network in Figure 2(a) and Figure 2(b). As shown in Figures 1 and 2, we are able to estimate the correct number of different types of communities when $\lambda \geq 0.65$ with $P'_{\text{BMCA}}(k_a = 4 \mid A) = 0.6$ $(P'_{\text{BMCA}}(k_b = 4 \mid A) = 0.6)$; then, when $\lambda \geq 0.8$, the proposal probability of the correct number



(a)

(b)

FIGURE 3: Test of BMCA against the projection method (PM) on synthetic networks generated using the bipartite stochastic block model in a difficult case. Posterior probabilities $P'\ (k_a = 2 \mid A)$ (a) and $P'\ (k_b = 3 \mid A)$ (b) are functions of the mixing parameter λ.

of communities $P'_{\mathrm{BMCA}}(k_a = 4 \mid A)$ and $P'_{\mathrm{BMCA}}(k_b = 4 \mid A)$ is equal to 1. However, the projection method gives poorer results (see the red lines in Figure 1), and it begins to estimate the correct number of different types of communities at $\lambda = 0.6$ with $P'_{\mathrm{PM}}(k_a = 4 \mid A) = 0.1$ ($P'_{\mathrm{PM}}(k_b = 4 \mid A) = 0.1$) and when $\lambda = 1$ $P'_{\mathrm{PM}}(k_a = 4 \mid A) = 0.9$ ($P'_{\mathrm{PM}}(k_b = 4 \mid A) = 0.8$). When $\lambda \geq 0.5$, $P'_{\mathrm{BMCA}}(k_a = 4 \mid A)$ ($P'_{\mathrm{BMCA}}(k_b = 4 \mid A)$) is always bigger than $P'_{\mathrm{PM}}(k_a = 4 \mid A)(P'_{\mathrm{PM}}(k_b = 4 \mid A))$, as shown in Figure 3.

4.2. A Difficult Case. In the difficult case, the synthetic networks are created with two type-a communities ($k_a = 2$) and three type-b communities ($k_a = 3$), and the degree of each node is different (i.e., the network with planted community structure has a heterogeneous degree distribution). The communities are set with different sizes, and we divide 700 type-a nodes evenly into 2 communities $\{350, 350\}$ and 300 type-b nodes into 3 communities $\{100, 150, 150\}$. Let $m_{1,3} = m_{2,4} = 2500$ and $m_{1,5} = m_{2,5} = 1500$; i.e., the total number of edges is 8000. Then, the block structure matrix $\omega^{\mathrm{planted}}$ can be defined with $\omega^{\mathrm{Planted}}_{1,3} = \omega^{\mathrm{Planted}}_{2,4} = 2500$ and $\omega^{\mathrm{Planted}}_{1,5} = \omega^{\mathrm{Planted}}_{2,5} = 1500$ (the symmetric entry has the same value), and the random network matrix ω^{random} can be defined with $\omega^{\mathrm{random}}_{1,3} = \omega^{\mathrm{random}}_{1,4} = \omega^{\mathrm{random}}_{2,3} = \omega^{\mathrm{random}}_{2,4} = 1250$ and $\omega^{\mathrm{random}}_{1,5} = \omega^{\mathrm{random}}_{2,5} = 1500$. The symmetric entry has the same value. Finally, with the specification above, we create networks of the difficult case to test our algorithm (the code to create these networks can be downloaded from [27]).

As shown by the blue line in Figure 3, when the level of noise is decreased, BMCA begins to estimate the correct number of communities for the network in a difficult case when $\lambda = 0.5$, and the posterior probabilities $P'\ (k_a = 2 \mid A)$ and $P'\ (k_b = 3 \mid A)$ increase sharply after $\lambda \geq 0.55$. Especially, when $\lambda \geq 0.75$, the proposal probability $P'_{\mathrm{BMCA}}(k_a = 2 \mid A) = 1$ ($P'_{\mathrm{BMCA}}(k_b = 3 \mid A) = 1$); i.e., we are always able to determine the correct number of communities. We used our method to calculate the probability of the number of communities in the synthetic mixing networks generated with $\lambda = 0.55$ and $\lambda = 0.6$, indicated as red circles in Figure 3, and show the result in Figure 4. As seen from Figures 3 and 4, our method can estimate the correct number of communities $k_a = 2$ and $k_b = 3$ in the synthetic network when $\lambda \geq 0.6$ for the difficult case. However, the projection method fails to estimate the correct number of communities even when $\lambda = 1$ and there is no noise, as shown by the red line in Figure 3.

4.3. Further Testing. We tested our method on synthetic networks of two different sizes of community as the number of network communities increases. The networks were generated using the bipartite stochastic block model with

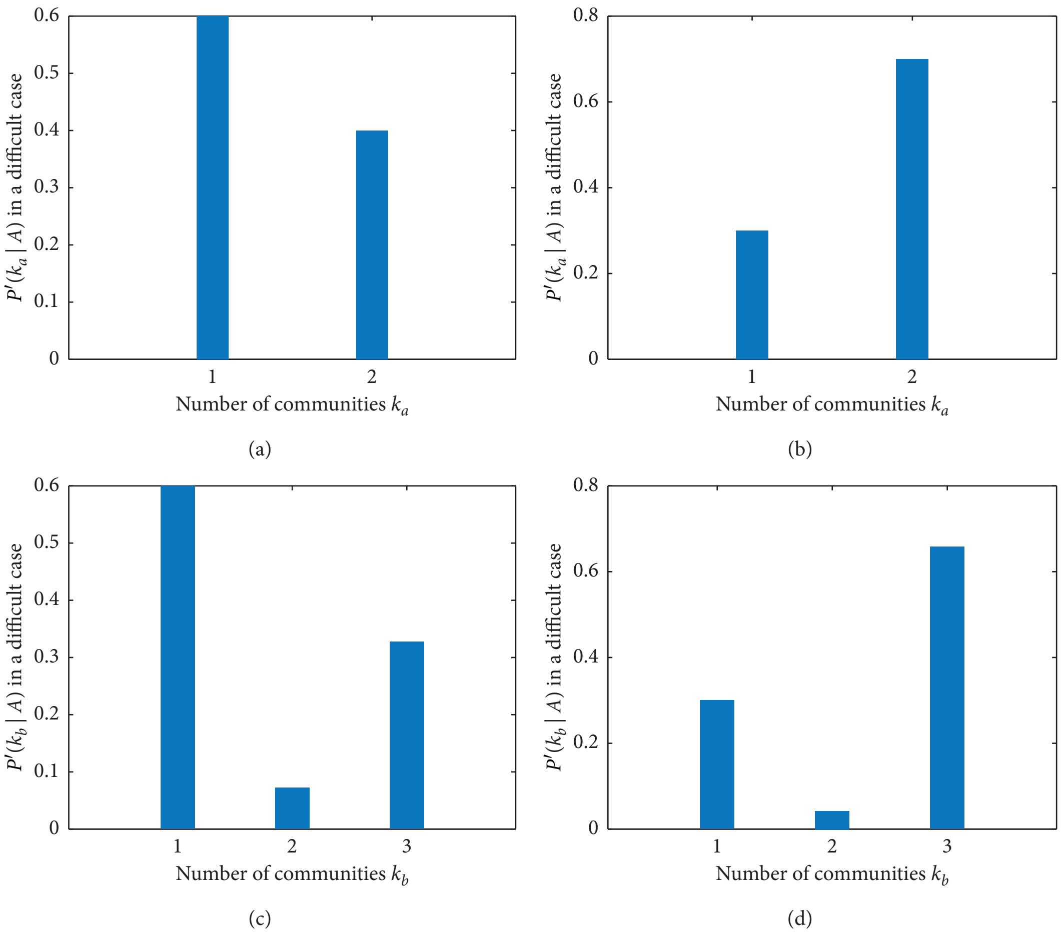


Figure 4: Posterior probabilities of the number of communities calculated for the synthetic networks generated using the bipartite stochastic block model in a difficult case. Posterior probabilities $P'(k_a \mid A)$ of the number of type-a communities in the synthetic networks generated with $\lambda = 0.55$ (a) and $\lambda = 0.55$ (b) and $P'(k_b \mid A)$ of the number of type-b communities in the synthetic networks generated with $\lambda = 0.55$ (c) and $\lambda = 0.6$ (d).

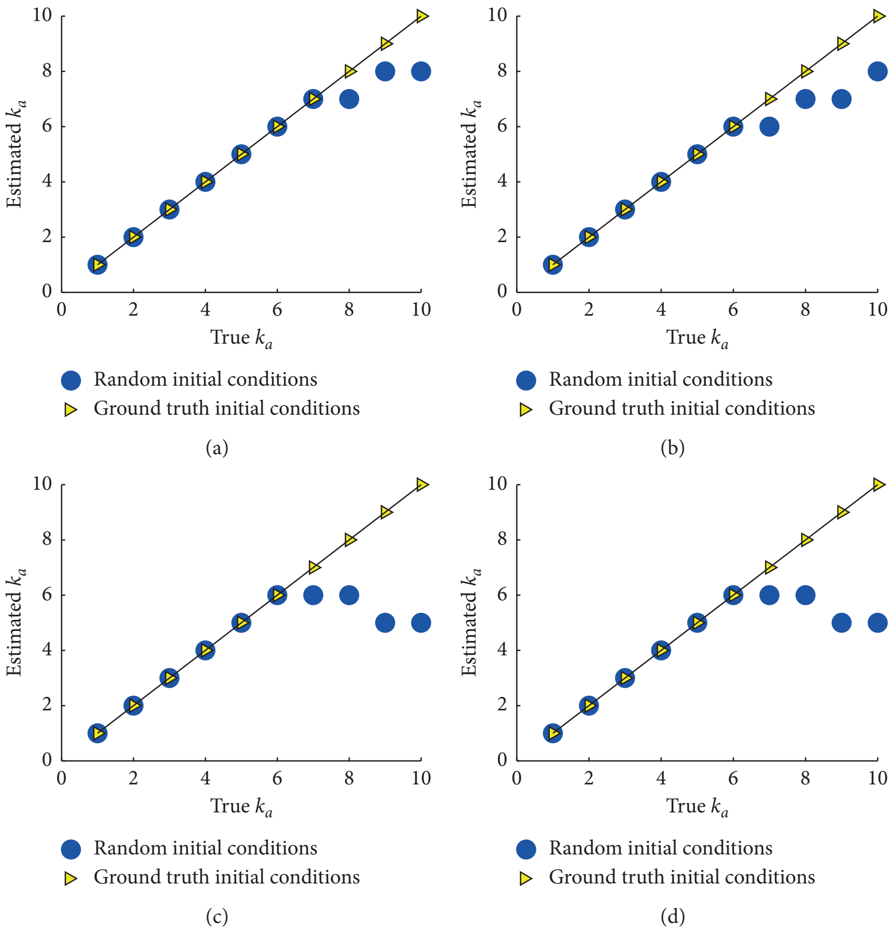
Table 2: The parameters set for synthetic networks.

Figure no.	$n_r\,(r = 1,\ldots,k)$	$\theta_i\,(i = 1,\ldots,N)$	$\omega_{rs}, r = 1,\ldots,k_a s = r + k_a$ and $k_a = 1,\ldots,10$
Figure 5(a)	250	Homogeneous	2500
Figure 5(b)	250	Heterogeneous	2500
Figure 5(c)	500	Homogeneous	5000
Figure 5(d)	500	Heterogeneous	5000

$\lambda = 1$, and the other parameters are set as listed in Table 2, where $\theta(i = 1,\ldots,N)$ is heterogeneous as real-world networks and the mean node degree of the network for the figures is 10. The number of communities k_a or k_b estimated using BMCA was correct until the actual number of communities increased to about 7, which is about 14 for k. The results are shown in Figure 5.

The results show a tendency to underestimate the number of communities for higher actual numbers of communities, especially when the size of the community changes from 250 (Figures 5(a) and 5(b)) to 500 (Figures 5(c) and 5(d)) and that of the networks increase correspondingly. However, these calculations appear when BMCA is run with a random initialization partition of nodes to communities. When BMCA is started on the same network with community partitions corresponding exactly to the planted community structure (yellow triangles), we always find the accurate number of communities.

Even so, the underestimation of k_a (or k_b) occurs not because the correct community partition fails to maximize the posterior probability $P(k_a \mid A)$ but rather because BMCA has not run for long enough to find the maximum. The method is theoretically sound, but when the number of possible community partitions k^N increases very rapidly with k, the numerical calculation becomes too demanding [4]. We can possibly design a more efficient Monte Carlo algorithm to



(a)

(b)

(c)

(d)

Figure 5: Tests of our algorithm on synthetic bipartite networks. In each subgraph, triangles represent results derived from Monte Carlo runs with the known correct partitions (ground truth initial conditions), while circles represent runs started with a random initial partition of nodes to community (random initial conditions).

solve this problem although it offers some useful information of a lower bound on the number of communities in the network given by BMCA.

5. Conclusions

In our paper, a new projection-free Bayesian inference method for determining the number of pure-type communities in a bipartite network has been introduced. First, we present the first principle derivation of a practical method, using the degree-corrected bipartite stochastic block model that is able to deal with networks with broad degree distributions, for estimating the number of pure-type communities of bipartite networks. Second, we propose a prior probability distribution over the partition of a bipartite network, with type parameter T ensuring that each community is pure type. Third, we design a Monte Carlo algorithm incorporated with our proposed method and prior probability distribution. We have illustrated the performance of the method with applications to a wide range of synthetic bipartite networks, including an easy case with homogeneous degree distributions and a difficult case with heterogeneous degree distributions. The results show that the proposed algorithm can determine the correct number of communities and perform better than the projection method, especially in networks with heterogeneous degree distributions.

However, our method underestimates the number of communities when the number of communities becomes large. The reason for this is due to the number of possible community partitions increasing very rapidly with the increase in the number of communities and because our algorithm has not run for long enough to find the posterior probability. Thus, our future work will focus on (i) finding a method that can more efficiently sample the posterior distribution over community partitions to correctly estimate a large number of communities in the network and (ii) extending applications on real-world data sets.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request. Our code and test data are available at http://my.shu.edu.cn/Web_LWZZ.aspx?TID=2887

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the Shanghai Municipal Education Commission of China (no. 2019PD1-2-37) and the Shanghai Youth Top-Notch Talent Development Program.

References

[1] M. E. J. Newman, "The structure of scientific collaboration networks," *SIAM Review*, vol. 45, pp. 167–256, 2013.

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

[3] E. Abbe, "Community detection and stochastic block models," *Foundations and Trends in Communications and Information Theory*, vol. 14, no. 1-2, pp. 1–162, 2018.

[4] M. A. Riolo, G. T. Cantwell, G. Reinert, and M. E. Newman, "Efficient method for estimating the number of communities in a network," *Physical Review E*, vol. 96, no. 3, 2017.

[5] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, 2006.

[6] D. B. Larremore, A. Clauset, and A. Z. Jacobs, "Efficiently inferring community structure in bipartite networks," *Physical Review E*, vol. 90, no. 1, 2014.

[7] Z. Razaee, A. Amini, Arash, and J. J. Li, "Matched bipartite block model with covariates," *Journal of Machine Learning Research*, vol. 20, pp. 1–44, 2019.

[8] M. Gerlach, T. P. Peixoto, and E. G. Altmann, "A network approach to topic models," *Science Advances*, vol. 4, no. 7, 2018.

[9] L. Feng, Q. C. Zhou, and Q. Zhao, "A spectral method to find communities in bipartite networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 513, pp. 424–437, 2019.

[10] M. J. Barber, "Modularity and community detection in bipartite networks," *Physical Review E*, vol. 76, no. 6, 2007.

[11] D. Melamed, "Community structures in bipartite networks: a dual-projection approach," *PLoS One*, vol. 9, Article ID e97823, 2014.

[12] S. J. Beckett, "Improved community detection in weighted bipartite networks," *Royal Society Open Science*, vol. 3, Article ID 140536, 2016.

[13] L. Xiong, G.-Z. Wang, and H.-C. Liu, "New community estimation method in bipartite networks based on quality of filtering coefficient," *Scientific Programming*, vol. 2019, Article ID 4310561, 12 pages, 2019.

[14] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, 2004.

[15] A. Miyauchi and N. Sukegawa, "Maximizing Barber's bipartite modularity is also hard," *Optimization Letters*, vol. 9, no. 5, pp. 897–913, 2014.

[16] S. Fortunato and D. Hric, "Community detection in networks: a user guide," *Physics Reports*, vol. 659, pp. 1–44, 2016.

[17] Z. Li, R.-S. Wang, S. Zhang, and X.-S. Zhang, "Quantitative function and algorithm for community detection in bipartite networks," *Information Sciences*, vol. 367–368, pp. 874–889, 2016.

[18] P. Pesantez-Cabrera and A. Kalyanaraman, "Efficient detection of communities in biological bipartite networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 1, pp. 258–271, 2019.

[19] P. J. Bickel and A. Chen, "A nonparametric view of network models and newman-girvan and other modularities," *Proceedings of the National Academy of Sciences*, vol. 106, no. 50, pp. 21068–21073, 2009.

[20] B. Karrer and M. E. J. Newman, "Stochastic block models and community structure in networks," *Physical Review E*, vol. 83, no. 1, 2011.

[21] T. P. Peixoto, "Hierarchical block structures and high-resolution model selection in large networks," *Physical Review X*, vol. 4, no. 1, 2014.

[22] A. Coja-Oghlan and A. Lanka, "Finding planted partitions in random graphs with general degree distributions," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 4, pp. 1682–1714, 2010.

[23] E. Côme and P. Latouche, "Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood," *Statistical Modelling: An International Journal*, vol. 15, no. 6, pp. 564–589, 2015.

[24] M. E. J. Newman and G. Reinert, "Estimating the number of communities in a network," *Physical Review Letters*, vol. 117, no. 7, 2016.

[25] M. Kastner, "Monte Carlo methods in statistical physics: mathematical foundations and strategies," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 6, pp. 1589–1602, 2010.

[26] M. E. J. Newman, "Network structure from rich but noisy data," *Nature Physics*, vol. 14, no. 6, pp. 542–545, 2018.

[27] biSBM, https://danlarremore.com/bipartiteSBM.