Network Structure Change Point Detection by Posterior Predictive Discrepancy

Lingbin Bian, Tiangang Cui, Georgy Sofronov and Jonathan Keith

Abstract Detecting changes in network structure is important for research into systems as diverse as financial trading networks, social networks and brain connectivity. Here we present novel Bayesian methods for detecting network structure change points. We use the stochastic block model to quantify the likelihood of a network structure and develop a score we call *posterior predictive discrepancy* based on sliding windows to evaluate the model fitness to the data. The parameter space for this model includes unknown latent label vectors assigning network nodes to interacting communities. Monte Carlo techniques based on Gibbs sampling are used to efficiently sample the posterior distributions over this parameter space.

Keywords Bayesian inference · Networks · Sliding window · Stochastic block model · Gibbs sampling

1 Introduction

Time varying network models are used in a wide range of applications, including in neuroscience where they have been used to model functional connectivity of brains such as the modularity models in [1, 2], and to model interactions in social network communities such as Facebook or emails [3]. The detection of changes in commu-

L. Bian (✉) · T. Cui · J. Keith
Monash University, 9 Rainforest Walk, Melbourne, VIC 3800, Australia
e-mail: lingbin.bian@monash.edu

T. Cui
e-mail: tiangang.cui@monash.edu

J. Keith
e-mail: jonathan.keith@monash.edu

G. Sofronov
Macquarie University, 12 Wally's Walk, Sydney, NSW 2109, Australia
e-mail: georgy.sofronov@mq.edu.au

B. Tuffin and P. L'Ecuyer (eds.), *Monte Carlo and Quasi-Monte Carlo Methods*,
Springer Proceedings in Mathematics & Statistics 324,
https://doi.org/10.1007/978-3-030-43465-6_5

nities, or more specifically changes in how nodes are allocated to communities, is important to understand functional variation in networks.

There is a wide range of literature exploring network change point analysis in time series. A recent method of network change point detection [4] used *spectral clustering* to partition a network into several connected components. The network structure deviance before and after the candidate change point was evaluated by computing the principal angles between two eigenspaces. The location of the change point was determined in such a way as to minimise a sum of singular values. Another method of network change point analysis named *dynamic connectivity regression* (DCR) [5, 6] used graphical LASSO (GLASSO) [7] to estimate a sparse precision matrix using an L_1-constraint, which forces a large number of edge weights to zero to represent missing edges. Both spectral clustering and DCR were integrated into the random permutation procedure [8] and stationary bootstrap procedure [9] to check whether detected change points were significant. Various criteria have been proposed as test scores to identify candidate change points in network connectivity, including summation of singular values of the network eigenspace (using spectral clustering as mentioned above) and the Bayesian information criterion (BIC) [6] in the context of dynamic connectivity regression. The BIC is a criterion for model selection that includes a penalty term for the number of parameters in the model, the implementation of which is illustrated in [10]. Apart from the greedy algorithm scheme in [5], a frequency-specific method described in [11] applied a multivariate cumulative sum procedure to detect change points. Some methods such as [12–14] mainly focused on large scale network estimation in time series. There are many papers using sliding window methods for observing the time varying network connectivity in time series analysis. For example, [15] tested the equality of the two covariance matrices in a high-dimensional setup within a sliding window to evaluate changes of connectivity in networks. Some other sliding window methods for network connectivity analysis can be found in [16–19]. Detection of communities in networks is also a relevant and topical area of statistics. How communities change or how the nodes in a network are assigned to specific communities is an important problem in characterization of networks. Theory and methods for community detection in networks are described in the works [20–22].

In this paper, we propose a new method to detect network structure change points using Bayesian model fitness assessment. There is a substantial literature on model fitness [23]. For example, West [24] used the cumulative Bayes factor to check for model failure, and Gelman [25] used posterior predictive assessment with a parameter dependent statistic to evaluate model fitness. In this work, we identify change points via checking model fitness to observations within a sliding time window using parameter dependent posterior predictive assessment. Specifically, we propose to use the stochastic block model [21, 26, 27] to quantify the likelihood of a network and Gibbs sampling to sample a posterior distribution derived from this model. The Gibbs sampling approach we adopt is based on the work of Nobile [28] for finite mixture models. We propose a posterior predictive discrepancy method to check model fitness using an adjacency matrix to represent a network. The proposed procedure involves drawing parameters from the posterior distribution and using them to generate a

replicated adjacency matrix, then calculating a *disagreement matrix* to quantify the difference between the replicated adjacency matrix and realised adjacency matrix. The score *posterior predictive discrepancy* (PPD) or we call the *posterior predictive discrepancy index* (PPDI) is then evaluated by averaging the fraction of elements in the disagreement matrix that indicate disagreement. We apply another new sliding window to construct a new time series we call the *cumulative discrepancy energy* (CDE). We compute the CDE and use it to define the criterion for change point detection. The CDE increases when change points are contained within the window, and can thus be used to assess whether a statistically significant change point exists within a period of time.

This paper is organized as the follows. Section 2 describes the details of the data time series, and illustrates the models and methodologies we propose for network change point detection. Section 3 contains results of numerical experiments and simulations. Section 4 assesses the advantages and disadvantages of our methods and potential future extensions and improvements.

2 Methods

2.1 The Data Set and Sliding Window Processing

Graphical models is a pictorial representation of pair-wise statistical relations between random variables. Graphical models may involve directed or undirected graphs. Directed graphs are appropriate when the nature of the relationships between variables has a directional aspect, whereas undirected graphs are appropriate for representing bi-directional or non-directional relationships. The methods we developed in this paper apply to both directed and undirected networks.

Consider a collection of N nodes $V = \{v_1, \ldots, v_N\}$. Suppose we observe a collection of N time series $\mathbf{Y} \in \Re^{N \times T}$ where $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T)$, with one time series corresponding to each node, and observations made at times $\{1, \ldots, T\}$. Correlations between time series indicate direct or indirect interactions between the corresponding nodes; we therefore first process the time series to construct a sequence of graphs in which edges represent temporary correlations.

We apply a sliding window technique with window length W which is considered to be an even number. The window size should be as small as possible. Large window size will limit the detection performance for those change points located closely with each other, while small window size may create statistical complication in the model assessment due to the lack of data sample. Change points may occur only at times $t \in \{M + 1, \ldots, T - M\}$ where M is a margin size used to avoid computational and statistical complications. We set the margin size $M = W/2$. For each time point $t \in \{M + 1, \ldots, T - M\}$, we define $\mathbf{Y}_t = \{\mathbf{y}_{t-\frac{W}{2}}, \ldots, \mathbf{y}_t, , \ldots, \mathbf{y}_{t+\frac{W}{2}-1}\}$ and calculate a sample correlation matrix $\mathbf{R}_t$ within the window $\mathbf{Y}_t$. We set a threshold ε such that only those node pairs (i, j) for which the correlation coefficient $r_{ij}^{(t)} > \varepsilon$ are

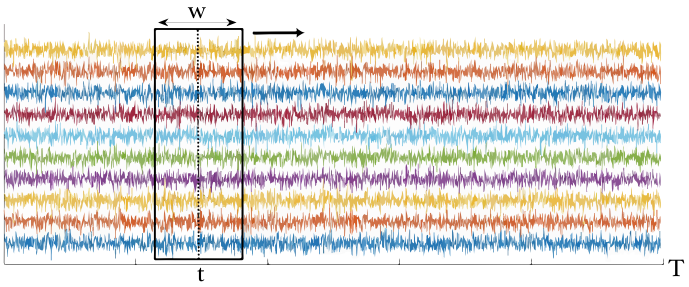

Fig. 1 Parallel time series corresponding to nodes of the network, and a sliding window of width W centred at t. The different coloured time series correspond to signal data for each node

connected by an edge in the edge set E_t representing interacting nodes at time t. It is also convenient to define an *adjacency matrix* $\mathbf{x}_t = (x_{ij}^{(t)})_{i,j=1,\ldots,N}$, where $x_{ij}^{(t)} = 1$ if there is an edge connecting nodes i and j in E_t, and $x_{ij}^{(t)} = 0$ otherwise. For each t, we then have the corresponding sample adjacency matrix $\mathbf{x}_t$ representing interacting nodes during the time window centred at time t. In what follows, we discard the signal data consisting of N time series, and instead consider the sample adjacency matrix $\mathbf{x}_t$ as the realised observation at time t. This sliding window approach is illustrated in Fig. 1.

2.2 The Stochastic Block Model

The stochastic block model is a random process generating networks on a fixed number N of nodes. A defining feature of the model is that nodes are partitioned into K communities, with interactions between nodes in the same community having a different (usually higher) probability than interactions between nodes in different communities. Taking a Bayesian perspective, we suppose that the number of communities K is a random variable drawn from a given prior distribution (for example a Poisson distribution). Determining the value of K appropriate to a given data set is a model selection problem. The stochastic block model first assigns the N nodes into the K communities, then generates edges with a probability determined by the community structure. Mathematically, we denote the *community memberships* (also called the *latent labels*) of the nodes as a random vector $\mathbf{z} = (z_1, \ldots, z_N)$ such that $z_i \in \{1, \ldots, K\}$ denotes the community containing node i. Each z_i independently follows categorical (one trial multinomial) distribution:

$$z_i \sim \text{Categorical}(1; r_1, \ldots, r_K),$$

where r_k is the probability of a node being assigned to community k and $\sum_{k=1}^{K} r_k = 1$. The multinomial probability can be expressed as

$$p(z_i|\mathbf{r}, K) = \prod_{k=1}^{K} r_k^{I_k(z_i)},$$

with the indicator function

$$I_k(z_i) = \begin{cases} 1, & \text{if } z_i = k \\ 0, & \text{if } z_i \neq k. \end{cases}$$

This implies that the N dimensional vector $\mathbf{z}$ is generated with probability

$$p(\mathbf{z}|\mathbf{r}, K) = \prod_{k=1}^{K} r_k^{m_k(\mathbf{z})},$$

where $m_k(\mathbf{z}) = \sum_{i=1}^{N} I_k(z_i)$. The vector $\mathbf{r} = (r_1, \ldots, r_K)$ is assumed to have a K-dimensional Dirichlet prior with density

$$p(\mathbf{r}|K) = N(\boldsymbol{\alpha}) \prod_{k=1}^{K} r_k^{\alpha_k - 1},$$

where the normalization factor with gamma function Γ is

$$N(\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)}.$$

In this work we suppose $\alpha_i = 1$ for $i = 1, \ldots, K$, so that the prior for $\mathbf{r}$ is uniform on the K-simplex.

Edges between nodes are represented using an adjacency matrix $\mathbf{x} \in \Re^{N \times N}$. Edges can be weighted or unweighted, and x_{ij} can be continuous or discrete. Here we use the *binary edge model*, in which $x_{ij} = 1$ for edges deemed present and $x_{ij} = 0$ for edges deemed absent. We define a *block* $\mathbf{x}_{kl}$ as the sub-matrix of the adjacency matrix comprised of edges connecting the nodes in community k to the nodes in community l. If the graph is undirected, there are $\frac{1}{2}K(K+1)$ blocks. If the graph is directed, there are K^2 blocks.

In the Bayesian presentation of the stochastic block model by MacDaid et al. [26], the likelihood model for edges is given by:

$$p(\mathbf{x}|\boldsymbol{\pi}, \mathbf{z}, K) = \prod_{k.l} p(\mathbf{x}_{kl}|\pi_{kl}, \mathbf{z}, K)$$

and

$$p(\mathbf{x}_{kl}|\pi_{kl}, \mathbf{z}, K) = \prod_{\{i|z_i=k\}} \prod_{\{j|z_j=l\}} p(x_{ij}|\pi_{kl}, \mathbf{z}, K)$$

where $\boldsymbol{\pi} = \{\pi_{kl}\}$ is a $K \times K$ matrix. In the binary edge model, each x_{ij} has a Bernoulli distribution, that is

$$x_{ij}|\pi_{kl}, \mathbf{z}, K \sim \text{Bernoulli}(\pi_{kl}).$$

The π_{kl} independently follow the conjugate Beta prior $\pi_{kl} \sim \text{Beta}(a, b)$. Let $n_{kl}(\mathbf{z}, \mathbf{x})$ be the number of edges in block kl (for the weighted edge model, n_{kl} becomes the sum of the edge weights). For an undirected graph, the number of edges connecting community k and community l is $n_{kl}(\mathbf{z}, \mathbf{x}) = \sum_{i,j|i\leq j, z_i=k, z_j=l} x_{ij}$. For a directed graph, $n_{kl}(\mathbf{z}, \mathbf{x}) = \sum_{i,j|z_i=k, z_j=l} x_{ij}$. We also define $w_{kl}(\mathbf{z})$ to be the maximum possible number of edges in block kl. For the off-diagonal blocks, $w_{kl}(\mathbf{z}) = m_k(\mathbf{z})m_l(\mathbf{z})$. For the diagonal blocks, if the graph is undirected, $w_{kk} = \frac{1}{2}m_k(\mathbf{z})(m_k(\mathbf{z}) + 1)$ (we consider the self-loop here), whereas if the graph is directed, $w_{kk} = m_k(\mathbf{z})^2$. With this notation, the probability associated with the edges of the block $\mathbf{x}_{kl}$ under the binary edge model is

$$p(\mathbf{x}_{kl}|\pi_{kl}, \mathbf{z}, K) = \pi_{kl}^{n_{kl}(\mathbf{z},\mathbf{x})}(1 - \pi_{kl})^{w_{kl}(\mathbf{z}) - n_{kl}(\mathbf{z},\mathbf{x})}, \text{ where } 0 < \pi_{kl} < 1.$$

The corresponding conjugate prior is the Beta distribution,

$$\text{Beta}(a, b) = \frac{\pi_{kl}^{a-1}(1 - \pi_{kl})^{b-1}}{B(a, b)},$$

where $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ is the Beta function.

2.3 The Collapsed Posterior

In the change point detection applications that we consider here, a change point corresponds to a restructuring of the network, that is, a change in the clustering vector $\mathbf{z}$. We are therefore interested in the so called "collapsed" posterior distribution $p(\mathbf{z}|\mathbf{x}, K)$, the form of which we discuss in this section.

We consider K unknown and assign a Poisson random prior with the condition $K > 0$.

$$P(K) = \frac{\lambda^K}{K!}e^{-\lambda}.$$

(In practice we use $\lambda = 1$.) We then have the joint density

$$p(\mathbf{x}, \boldsymbol{\pi}, \mathbf{z}, \mathbf{r}, K) = P(K)p(\mathbf{z}, \mathbf{r}|K)p(\mathbf{x}, \boldsymbol{\pi}|\mathbf{z}).$$

The parameters $\mathbf{r}$ and $\boldsymbol{\pi}$ can be integrated out or "collapsed" to obtain the marginal density $p(\mathbf{x}, \mathbf{z}, K)$.

$$p(\mathbf{z}, K, \mathbf{x}) = P(K) \int p(\mathbf{z}, \mathbf{r}|K) d\mathbf{r} \int p(\mathbf{x}, \boldsymbol{\pi}|\mathbf{z}) d\boldsymbol{\pi},$$

then the posterior for the block-wise model can be expressed as

$$p(\mathbf{z}, K|\mathbf{x}) \propto p(\mathbf{z}, K, \mathbf{x}) = P(K) \int p(\mathbf{z}, \mathbf{r}|K) d\mathbf{r} \prod_{k,l} \int p(\mathbf{x}_{kl}, \pi_{kl}|\mathbf{z}) d\pi_{kl}.$$

The first integral

$$p(\mathbf{z}|K) = \int p(\mathbf{z}, \mathbf{r}|K) d\mathbf{r},$$

where the integral is over the K-simplex, can be calculated via the following procedure:

$$\begin{aligned} p(\mathbf{z}|K) &= \int p(\mathbf{z}, \mathbf{r}|K) d\mathbf{r} \\ &= \int p(\mathbf{r}|K) p(\mathbf{z}|\mathbf{r}, K) d\mathbf{r} \\ &= \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\Gamma(\sum_{k=1}^{K} (\alpha_k + m_k(\mathbf{z}))} \prod_{k=1}^{K} \frac{\Gamma(\alpha_k + m_k(\mathbf{z}))}{\Gamma(\alpha_k)}. \end{aligned}$$

Integrals of the form $\int p(\mathbf{x}_{kl}, \pi_{kl}|\mathbf{z}) d\pi_{kl}$ can be calculated as

$$\begin{aligned} p(\mathbf{x}_{kl}|\mathbf{z}) &= \int_0^1 p(\mathbf{x}_{kl}, \pi_{kl}|\mathbf{z}) d\pi_{kl} \\ &= \int_0^1 p(\pi_{kl}) p(\mathbf{x}_{kl}|\pi_{kl}, \mathbf{z}) d\pi_{kl} \\ &= \frac{B(n_{kl}(\mathbf{z}, \mathbf{x}) + a, w_{kl}(\mathbf{z}) - n_{kl}(\mathbf{z}, \mathbf{x}) + b)}{B(a, b)}. \end{aligned}$$

The derivation of the collapsing procedure is given in Appendix "Derivation of the Collapsing Procedure". Then the collapsed posterior can be expressed as

$$p(\mathbf{z}|\mathbf{x}, K) \propto \frac{1}{K!} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\Gamma(\sum_{k=1}^{K} (\alpha_k + m_k))} \prod_{k=1}^{K} \frac{\Gamma(\alpha_k + m_k)}{\Gamma(\alpha_k)} \prod_{k,l} \frac{B(n_{kl} + a, w_{kl} - n_{kl} + b)}{B(a, b)}.$$

2.4 Sampling the Parameters from the Posterior

The posterior predictive method we outline below involves sampling parameters from the posterior distribution. The sampled parameters are the latent labels $\mathbf{z}$ and

model parameters $\boldsymbol{\pi}$. There are several methods for estimating the latent labels and model parameters of a stochastic block model described in the literature: for example Daudin et al. [29] evaluate the model parameters by point estimation but consider the latent labels in $\mathbf{z}$ as having a distribution, making their approach similar to an EM algorithm. The method of Zhangi et al. [30] uses point estimation for both the model parameters and latent labels. Here we sample the latent labels $\mathbf{z}$ from the collapsed posterior $p(\mathbf{z}|\mathbf{x}, K)$ and then separately sample $\boldsymbol{\pi}$ from the density $p(\boldsymbol{\pi}|\mathbf{x}, \mathbf{z})$.

The estimation of K is a model selection problem [26], which we will not discuss about in this paper. It is convenient to consider the number of communities K to be fixed in the model fitness assessment in this paper (The K is supposed to be given in the numerical experiment in the later section). We use the Gibbs sampler to sample the latent labels $\mathbf{z}$ from the collapsed posterior $p(\mathbf{z}|\mathbf{x}, K)$. For each element z_i and $k \in \{1, \ldots, K\}$, we have

$$p(z_i|z_{-i}, \mathbf{x}, K) = \frac{1}{C} p(z_1, \ldots, z_{i-1}, z_i = k, z_{i+1}, \ldots, z_n|\mathbf{x}),$$

where z_{-i} represents the elements in $\mathbf{z}$ apart from z_i and the normalization term

$$C = p(z_{-i}|\mathbf{x}, K) = \sum_{k=1}^{K} p(z_1, \ldots, z_{i-1}, z_i = k, z_{i+1}, \ldots, z_n|\mathbf{x}).$$

We use the standard Gibbs sampling strategy of cycling through $z_1, \ldots, z_n$, updating each latent variable by drawing from $p(z_i|z_{-i}, \mathbf{x}, K)$.

An alternative to Gibbs sampling is to use Metropolis-Hastings moves based on the allocation sampler [31] to draw parameters $\mathbf{z}$ and K from the posterior. In this approach, a candidate vector of latent labels $\mathbf{z}^*$ is accepted with probability $\min\{1, r\}$, where

$$r = \frac{p(K, \mathbf{z}^*, \mathbf{x}) p(\mathbf{z}^* \to \mathbf{z})}{p(K, \mathbf{z}, \mathbf{x}) p(\mathbf{z} \to \mathbf{z}^*)}.$$

If the number of communities K is fixed, the proposal $p(\mathbf{z} \to \mathbf{z}^*)$ can be based on three kinds of moves (M1, M2, M3). If K is allowed to vary, one can use a reversible jump strategy or absorption/ejection move. The details of these approaches are illustrated in [31, 33].

To sample the model parameters $\boldsymbol{\pi}$, we first derive the posterior of the model block parameters as the following expression

$$\begin{aligned} p(\pi_{kl}|\mathbf{x}_{kl}, \mathbf{z}) &\propto p(\pi_{kl}) p(\mathbf{x}_{kl}|\pi_{kl}, \mathbf{z}) \\ &\propto \pi_{kl}^{a-1}(1-\pi_{kl})^{b-1} \pi_{kl}^{n_{kl}(\mathbf{z},\mathbf{x})}(1-\pi_{kl})^{w_{kl}(\mathbf{z})-n_{kl}(\mathbf{z},\mathbf{x})} \\ &\propto \pi_{kl}^{n_{kl}(\mathbf{z},\mathbf{x})+a-1}(1-\pi_{kl})^{w_{kl}(\mathbf{z})-n_{kl}(\mathbf{z},\mathbf{x})+b-1} \end{aligned}$$

and

$$p(\boldsymbol{\pi}|\mathbf{x}, \mathbf{z}) = \prod_{k,l} p(\pi_{kl}|\mathbf{x}_{kl}, \mathbf{z}).$$

The prior and the likelihood in the above expression is the Beta-Bernoulli conjugate pair. Given the sampled $\mathbf{z}$ we can draw the sample $\boldsymbol{\pi}$ from the above posterior directly.

2.5 Posterior Predictive Discrepancy

Given inferred values of $\mathbf{z}$ and $\boldsymbol{\pi}$ under the assumptive model K, one can draw replicated data $\mathbf{x}^{rep}$ from the posterior predictive distribution $P(\mathbf{x}^{rep}|\mathbf{z}, \boldsymbol{\pi}, K)$. Note that the realised adjacency and replicated adjacency are conditionally independent,

$$P(\mathbf{x}, \mathbf{x}^{rep}|\mathbf{z}, \boldsymbol{\pi}, K) = P(\mathbf{x}^{rep}|\mathbf{z}, \boldsymbol{\pi}, K)P(\mathbf{x}|\mathbf{z}, \boldsymbol{\pi}, K).$$

Multiplying both sides of this equality by $P(\mathbf{z}, \boldsymbol{\pi}|\mathbf{x}, K)/P(\mathbf{x}|\mathbf{z}, \boldsymbol{\pi}, K)$ gives

$$P(\mathbf{x}^{rep}, \mathbf{z}, \boldsymbol{\pi}|\mathbf{x}, K) = P(\mathbf{x}^{rep}|\mathbf{z}, \boldsymbol{\pi}, K)P(\mathbf{z}, \boldsymbol{\pi}|\mathbf{x}, K).$$

Here we use replicated data in the context of posterior predictive assessment [25] to evaluate the fitness of a posited stochastic block model to a realised adjacency matrix. We generate a replicated adjacency matrix by first drawing samples $(\mathbf{z}, \boldsymbol{\pi})$ from the joint posterior $P(\mathbf{z}, \boldsymbol{\pi}|\mathbf{x}, K)$. Specifically, we sample the latent label vector $\mathbf{z}$ from $p(\mathbf{z}|\mathbf{x}, K)$ and model parameter $\boldsymbol{\pi}$ from $p(\boldsymbol{\pi}|\mathbf{x}, \mathbf{z})$ and then draw a replicated adjacency matrix from $P(\mathbf{x}^{rep}|\mathbf{z}, \boldsymbol{\pi}, K)$. We compute a discrepancy function to assess the difference between the replicated data $\mathbf{x}^{rep}$ and the realised observation $\mathbf{x}$, as a measure of model fitness.

In [25], the χ^2 function was used as the discrepancy measure, where the observation was considered as a vector. However, in the stochastic block model, the observation is an adjacency matrix and the sizes of the sub-matrices can vary. In this paper, we propose a *disagreement index* to compare binary adjacency matrices $\mathbf{x}^{rep}$ and $\mathbf{x}$. We use the exclusive OR operator to compute the disagreement matrix between the realised adjacency and replicated adjacency and calculate the fraction of non-zero elements in the disagreement matrix. This disagreement index is denoted $\gamma(\mathbf{x}^{rep}; \mathbf{x})$ and can be considered a parameter-dependent statistic. In mathematical notation, the disagreement index γ is defined as

$$\gamma(\mathbf{x}^{rep}; \mathbf{x}) = \frac{\sum_{i=1,j=1}^{N} (\mathbf{x} \bigoplus \mathbf{x}^{rep})_{ij}}{N^2},$$

where $\bigoplus$ is the exclusive OR operator. In practice we generate S replicated adjacency matrices and compute the average disagreement index, we call *posterior predictive discrepancy index* (PPDI)

$$\overline{\gamma} = \frac{\sum_{i=1}^{S} \gamma(\mathbf{x}^{rep^i}; \mathbf{x})}{S}.$$

2.6 Cumulative Discrepancy Energy via Sliding Window

Our proposed strategy to detect network change points is to assess the fitness of a stochastic block model by computing the discrepancy index $\overline{\gamma}_t$ for each $t \in \{\frac{W}{2} + 1, \ldots, T - \frac{W}{2}\}$. The key insight here is that the fitness of the model is relatively worse when there is a change point within the window used to compute $\mathbf{x}_t$. If there is a change point within the window, the data observed in the left segment and right segment are generated by different network architectures, resulting in poor model fit and a correspondingly high posterior predictive discrepancy index.

We find that the PPDI is greatest when the change point is located in the middle of the window. To identify the most plausible position of a change point, we use another window with window size W_s to smooth the results. We compute the *cumulative discrepancy energy* E_t, given by

$$E_t = \sum_{i=t-\frac{W_s}{2}}^{t+\frac{W_s}{2}-1} \overline{\gamma}_i.$$

We infer the location of change points to be local maxima of the cumulative discrepancy energy, where those maxima rise sufficiently high above the surrounding sequence. The change point detection algorithm can be summarized as the follows.

Algorithm 1 Change point detection by posterior predictive discrepancy

Input: Length of time course T, window size W, number of communities K, observations $\mathbf{Y}$.

for $t = W/2 + 1, \ldots, T - W/2$ **do**

Calculate $\mathbf{Y}_t \to \mathbf{R}_t \to \mathbf{x}_t$.

Draw the samples $\{\mathbf{z}^i, \boldsymbol{\pi}^i\}$ $(i = 1, \ldots, S)$ from the posterior $P(\mathbf{z}, \boldsymbol{\pi} | \mathbf{x}, K)$.

Simulate the replicated set $\mathbf{x}^{rep^i}$ from the predictive distribution $P(\mathbf{x}^{rep} | \mathbf{z}, \boldsymbol{\pi}, K)$.

Calculate the disagreement index $\gamma(\mathbf{x}^{rep^i}; \mathbf{x})$.

Calculate the posterior predictive discrepancy index $\overline{\gamma}_t = \frac{1}{S} \sum_{i=1}^{S} \boldsymbol{\gamma}(\mathbf{x}^{rep^i}; \mathbf{x})$.

end for

for $t = \frac{W}{2} + \frac{W_s}{2} + 1, \ldots, T - \frac{W}{2} - \frac{W_s}{2}$ **do**

Calculate cumulative discrepancy energy $E_t = \sum_{I=t-\frac{W_s}{2}}^{t+\frac{W_s}{2}-1} \overline{\gamma}_I$.

end for

3 Simulation

3.1 Generative Model

To validate our approach, we simulate the time series consisting of three data segments from the Gaussian generative model. Within each of the resulting segment, $N = 16$ nodes are assigned to $K = 3$ communities, resulting in membership vectors $\mathbf{z}_1$, $\mathbf{z}_2$ and $\mathbf{z}_3$. Recall these are generated using the Dirichlet-Categorical conjugate pair, that is, component weights $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are first drawn from a uniform distribution on the K-simplex and then nodes are assigned to the communities by drawing from the corresponding categorical distributions. Time series data in $\Re^N$ are then simulated for $t = 1, \ldots, T$ by drawing from a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, with

$$\Sigma_{ij} = \begin{cases} a, & \text{if } i \neq j \text{ and } i \text{ and } j \text{ are in the same communities} \\ 1, & \text{if } i = j \\ b, & \text{if } i \text{ and } j \text{ are in different communities.} \end{cases}$$

In the covariance matrix, a and b follow the uniform distribution, where $a \sim U(0.8, 1)$ and $b \sim U(0, 0.2)$. The resulting covariance matrices for the three segments we denote by $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$ and $\boldsymbol{\Sigma}_3$. The simulated data $\mathbf{Y} \in \Re^{N \times T}$ can be separated into three segments $(\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3)$.

3.2 Effect of Changing the Distance Between Change Points

We simulate the time series for a network with $N = 16$ nodes and $T = 450$ time points with different locations of true change points in four experimental settings. The sliding window size is fixed to be $W = 64$ so that the margin size is $M = 32$.

For the inference, we set the prior of π_{kl} to be $Beta(2, 2)$. During the posterior predictive procedure, according to the convergence performance of the Gibbs sampler, the Gibbs chain of the latent label vectors converges to the stationary distribution within 10 iterations. Then we draw each latent label vector every three complete Gibbs iterations. The posterior prediction replication number S determines the rate of fluctuation of the posterior predictive discrepancy index (PPDI) curve, the smaller the replication number is, the more severely the curve will vibrate. In this demonstration, we set the replication number as $S = 50$. Increasing S would lead to more accurate results, but incur additional computational cost.

The PPDI increases dramatically when the true change point begins to appear at the right of the sliding window and decreases rapidly when the true change point tend to move out the left end of the window. For the cumulative discrepancy energy (CDE), the change point is considered to be at the place where the CDE is a local maximum.

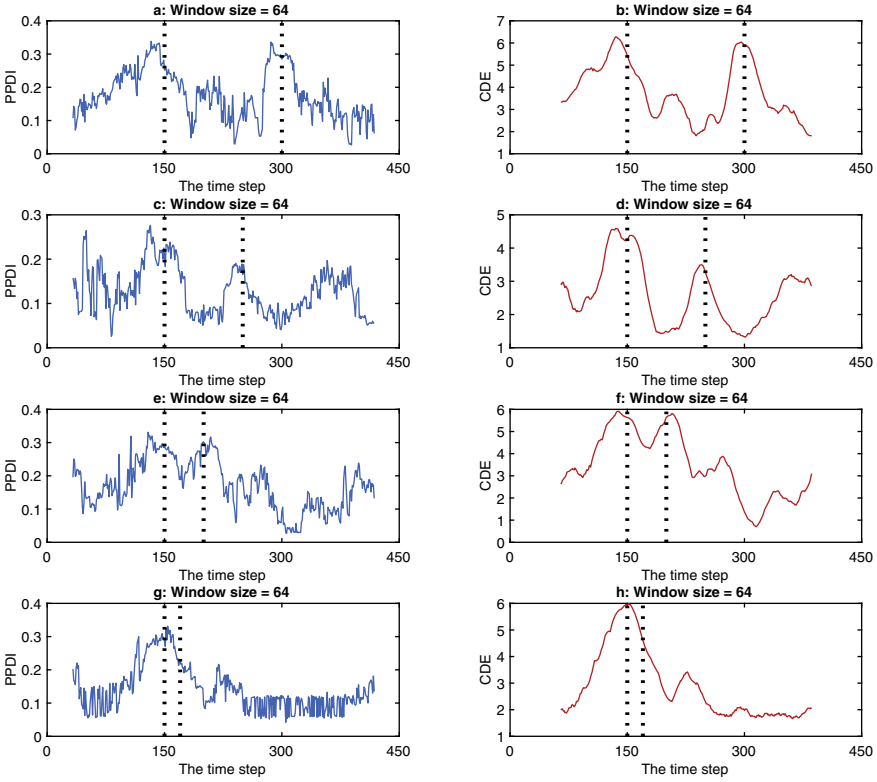

Fig. 2 The vertical lines in the figure represent the various locations of the true change points, the blue curve represents the posterior predictive discrepancy index (PPDI) and the red curve represents the cumulative discrepancy energy (CDE) with window size $W = 64$. **a** PPDI with change points at $t_1 = 150$ and $t_2 = 300$, **b** CDE with change points at $t_1 = 150$ and $t_2 = 300$; **c** PPDI with change points at $t_1 = 150$ and $t_2 = 250$, **d** CDE with change points at $t_1 = 150$ and $t_2 = 250$; **e** PPDI with change points at $t_1 = 150$ and $t_2 = 200$, **f** CDE with change points at $t_1 = 150$ and $t_2 = 200$; **g** PPDI with change points at $t_1 = 150$ and $t_2 = 170$, **h** CDE with change points at $t_1 = 150$ and $t_2 = 170$

In our first setting, true change points are placed at times $t_1 = 150$ and $t_2 = 300$ (see Fig. 2a, b). Note the minimum distance between change points is 150, which is larger than the window size. Consequently, no window can contain more than one change point. We can see from the figure that the two peaks are located around the true change points $t_1 = 150$ and $t_2 = 300$ respectively.

We repeat this experiment with the true change points at $t_1 = 150$ and $t_2 = 250$ in Fig. 2c, d so that the minimum distance between the change points is 100, which is still larger than the window size. We can see that there are two prominent peaks located around the true change points. Next, we set the true change points at $t_1 = 150$ and $t_2 = 200$ Fig. 2e, f, where the minimum distance between the change points is 50 which is slightly smaller than the window size 64. In this situation, the window may contain two change points, so that these windows cross three segments generated by different network architectures. We can still distinguish the two peaks in Fig. 2e, f

because the distance of the change points is still large enough. However, in Fig. 2g, h where the change points are $t_1 = 150$ and $t_2 = 170$, we can see that there are only one peak around $t = 150$. In this case, we cannot distinguish two change points because they are closely located with each other.

3.3 *Effect of Changing the Window Size*

To investigate the effect of changing the window size, we set the true change points at $t_1 = 150$ and $t_2 = 300$ for all of the experimental settings. We apply our method with four different window sizes: $W = 24$ in Fig. 3a, b; $W = 32$ in Fig. 3c, d; $W = 48$ in Fig. 3e, f; $W = 64$ in Fig. 3g, h. Reducing the window size will increase the fluctuation of the PPDI and CDE, and renders the change point locations less distinguishable. For $W = 24$, we can see that there are multiple large peaks over the CDE time series.

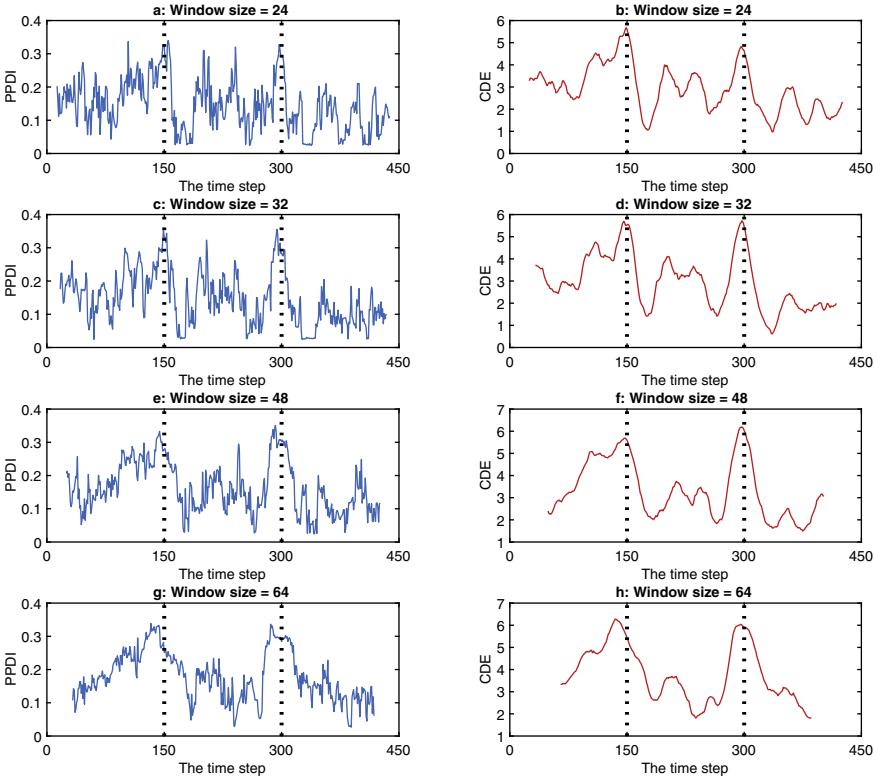

Fig. 3 The vertical lines in the figure represent the locations of the true change points, the blue curve represents the posterior predictive discrepancy index (PPDI) and the red curve represents the cumulative discrepancy energy (CDE) with window sizes 24, 32, 48 and 64

4 Discussion

The method for network structure change point detection described in this paper provides a flexible approach to modelling and estimating community structures among interacting nodes. We consider both the community latent label vector and block model parameters (block edge probabilities) as random variables to be estimated. Structural changes to the networks are reflected by changes in the latent labels and model parameters. By applying a sliding window method, we avoid partitioning the data into sub-segments recursively as in the algorithm of [4]. Compared to the method of evaluating eigen-structure of the network in [4], our approach has several advantages. Our approach is able to be used for both undirected and directed graphs. The method using stochastic block model is more flexible, because different choices of $\boldsymbol{\pi}$ can generate different connection patterns in the adjacency matrix. However, both of the methods have difficulty in detecting change points with close distances.

Ideally, the window size should be as small as possible, which can enhance the ability of detecting those change points located closely. When the window size is small, for example when $W = 24$, there may be false detections, because there are not enough samples of data in the sliding windows. In our current method, the window size cannot be made too small, which may be because we use a threshold to convert the sample correlation matrix into an adjacency matrix. This practice results in the loss of some information regarding the network architecture. If we extend the model to a weighted stochastic block model to fit the data in the future, so that the sample correlation matrix is directly considered as a weighted adjacency matrix, it may be feasible to detect change points at smaller separations and make the higher resolution of detecting the change points. For the majority of the applications in fMRI time series analysis, the time course should be around hundreds of time steps, which is because of the limitation of the sample time interval of the fMRI in the short time experiment. Therefore, the algorithm to analyse the short term network time series is important.

The computational cost of the posterior predictive discrepancy procedure in our method depends mainly on two aspects. The first includes the iterated Gibbs steps used to update the latent variables and the sampling of the model parameter. In our code, calculating $m(\mathbf{z})$ takes $O(N)$ time, calculating the probability of each element z_i to be reassigned into one of K clusters takes $O(K^2 + N^2 + KN)$ time. Therefore, iterating each latent vector $\mathbf{z}$ requires the computational cost of $O((K^2 + N^2 + KN)KN)$, sampling $\boldsymbol{\pi}$ requires $O(K^2 + N^2)$ time. The second is the number of replications needed for the predictive process. Calculating each PPDI requires $O(S)$ time. There is a natural trade off between increasing the replication number and reducing the computational cost.

In this paper, we have not considered the problem of inferring the number of communities K. In real world applications, K is unknown. Determination of K can be considered as a model selection problem, a class of problem for which many methods exist, including [34] in Bayesian statistics. For example, the allocation sampler [31] is an efficient tool for inference of K, and could potentially be integrated into our algorithm. In real word applications, some change points may not occur abruptly, but

rather change gradually over time. For solving the gradual changing problem, we may potentially apply a transition matrix to the latent label vectors in the generative model between difference segments to simulate the time series with ground truth of gradual change points. We do not claim that our Gibbs sampling approach is optimal, finding alternative sampling methods is thus another possibility for improving the algorithm. One idea that is worth exploring in the future is to develop efficient sampling methods for inferring high-dimensional latent vectors in larger scale networks.

5 Conclusion

The main contribution of this paper is to demonstrate that posterior predictive discrepancy criterion can be used to detect network structure change point based on time series data. This insight is potentially applicable to a wide range of applications including analysis of fMRI data and large scale social networks.

Acknowledgements The authors are grateful to the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers for their support of this project (CE140100049).

Appendix: Derivation of the Collapsing Procedure

We now create a new parameter vector $\boldsymbol{\eta} = \{\alpha_1 + m_1, \ldots, \alpha_K + m_K\}$. We can collapse the integral $\int p(\mathbf{z}, \mathbf{r}|K)d\mathbf{r}$ as the following procedure.

$$\begin{aligned}
\int p(\mathbf{z}, \mathbf{r}|K)d\mathbf{r} &= \int p(\mathbf{r}|K)\, p(\mathbf{z}|\mathbf{r}, K)d\mathbf{r} \\
&= \int N(\boldsymbol{\alpha}) \prod_{k=1}^{K} r_k^{\alpha_k - 1} \prod_{k=1}^{K} r_k^{m_k} d\mathbf{r} \\
&= \int N(\boldsymbol{\alpha}) \prod_{k=1}^{K} r_k^{\alpha_k + m_k - 1} d\mathbf{r} \\
&= \frac{N(\boldsymbol{\alpha})}{N(\boldsymbol{\eta})} \int N(\boldsymbol{\eta}) \prod_{k=1}^{K} r_k^{\alpha_k + m_k - 1} d\mathbf{r} \\
&= \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\Gamma(\sum_{k=1}^{K} (\alpha_k + m_k)} \prod_{k=1}^{K} \frac{\Gamma(\alpha_k + m_k)}{\Gamma(\alpha_k)}.
\end{aligned}$$

Integral of the form $\int p(\mathbf{x}_{kl}, \pi_{kl}|\mathbf{z})d\pi_{kl}$ can be calculated as

$$\begin{aligned}
\int_0^1 p(\mathbf{x}_{kl}, \pi_{kl}|\mathbf{z})d\pi_{kl} &= \int_0^1 p(\pi_{kl})\, p(\mathbf{x}_{kl}|\pi_{kl}, \mathbf{z})d\pi_{kl} \\
&= \int_0^1 \frac{\pi_{kl}^{a-1}(1-\pi_{kl})^{b-1}}{B(a,b)} \pi_{kl}^{n_{kl}}(1-\pi_{kl})^{w_{kl}-n_{kl}} d\pi_{kl} \\
&= \int_0^1 \frac{\pi_{kl}^{n_{kl}+a-1}(1-\pi_{kl})^{w_{kl}-n_{kl}+b-1}}{B(a,b)} d\pi_{kl} \\
&= \frac{B(n_{kl}+a, w_{kl}-n_{kl}+b)}{B(a,b)} \\
&\quad \times \int_0^1 \frac{\pi_{kl}^{n_{kl}+a-1}(1-\pi_{kl})^{w_{kl}-n_{kl}+b-1}}{B(n_{kl}+a, w_{kl}-n_{kl}+b)} d\pi_{kl} \\
&= \frac{B(n_{kl}+a, w_{kl}-n_{kl}+b)}{B(a,b)}.
\end{aligned}$$

References

1. Bassett, D.S., Porter, M.A., Wymbs, N.F., Grafton, S.T., Carlson, J.M., Mucha, P.J.: Robust detection of dynamic community structure in networks. CHAOS **23**, 013142 (2013)
2. Bassett, D.S., Wymbs, N.F., Porter, M.A., Mucha, P.J., Carlson, J.M., Grafton, S.T.: Dynamic reconfiguration of human brain networks during learning. PNAS **108**(18), 7641–7646 (2011)
3. Kawash, J., Agarwal, N., özyer, T.: Prediction and Inference from Social Networks and Social Media. Lecture Notes in Social Networks (2017)
4. Cribben, I., Yu, Y.: Estimating whole-brain dynamics by using spectral clustering. J. R. Stat. Soc., Ser. C (Appl. Stat.) **66**, 607–627 (2017)
5. Cribben, I., Haraldsdottir, R., Atlas, L.Y., Wager, T.D., Lindquist, M.A.: Dynamic connectivity regression: determining state-related changes in brain connectivity. NeuroImage **61**, 907–920 (2012)
6. Cribben, I., Wager, T.D., Lindquist, M.A.: Detecting functional connectivity change points for single-subject fMRI data. Front. Comput. Neurosci. **7**, 143 (2013)
7. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics **9**(3), 432–441 (2007)
8. Good, P.: Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses. Springer Series in Statistics (2000)
9. Politis, D.N., Romano, J.P.: The stationary bootstrap. J. Am. Stat. Assoc. **89**(428), 1303–1313 (1994)
10. Konishi, S., Kitagawa, G.: Information Criteria and Statistical Modeling. Springer Series in Statistics (2008)
11. Schröder, A.L., Ombao, H.: FreSpeD: frequency-specific change-point detection in epileptic seizure multi-channel EEG data. J. Am. Stat. Assoc. (2015)
12. Frick, K., Munk, A., Sieling, H.: Multiscale change point inference (with discussion). J. R. Stat. Society. Ser. B (Methodol.) **76**, 495–580 (2014)
13. Cho, H., Fryzlewicz, P.: Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. J. R. Stat. Society. Ser. B (Methodol.) **77**, 475–507 (2015)
14. Wang, T., Samworth, R.J.: High-dimensional change point estimation via sparse projection. J. R. Stat. Society. Ser. B (Methodol.) **80**(1), 57–83 (2017)

15. Jeong, S.-O., Pae, C., Park, H.-J.: Connectivity-based change point detection for large-size functional networks NeuroImage **143**, 353–363 (2016)
16. Chang, C., Glover, G.H.: Time-frequency dynamics of resting-state brain connectivity measured with fMRI. NeuroImage **50**, 81–98 (2010)
17. Handwerker, D.A., Roopchansingh, V., Gonzalez-Castillo, J., Bandettini, P.A.: Periodic changes in fMRI connectivity. NeuroImage **63**, 1712–1719 (2012)
18. Monti, R.P., Hellyer, P., Sharp, D., Leech, R., Anagnostopoulos, C., Montana, G.: Estimating time-varying brain connectivity networks from functional MRI time series. NeuroImage **103**, 427–443 (2014)
19. Allen, E.A., Damaraju, E., Plis, S.M., Erhardt, E.B., Eichele, T., Calhoun, V.D.: Tracking whole-brain connectivity dynamics in the resting state. Cereb. Cortex **24**, 663–676 (2014)
20. Newman, M.E.J.: Modularity and community structure in networks. PNAS **103**(23), 8577–8582 (2006)
21. Wang, Y.X.R., Bickel, P.J.: Likelihood-based model selection for stochastic block models. Ann. Stat. **45**(2), 500–528 (2017)
22. Jin, J.: Fast community detection by SCORE. Ann. Stat. **43**(1), 57–89 (2015)
23. Rubin, D.B.: Bayesianly justifiable and relevant frequency calculations for the applied statistician. Ann. Stat. **12**(4), 1151–1172 (1984)
24. West, M.: Bayesian model monitoring. J. R. Stat. Society. Ser. B (Methodol.) **48**(1), 70–78 (1986)
25. Gelman, A., Meng, X.-L., Stern, H.: Posterior predictive assessment of model fitness via realised discrepancies. Stat. Sin. **6**, 733–807 (1996)
26. MacDaid, A.F., Murphy, T.B., Friel, N., Hurley, N.J.: Improved Bayesian inference for the stochastic block model with application to large networks. Comput. Stat. Data Anal. **60**, 12–31 (2012)
27. Ridder, S.D., Vandermarliere, B., Ryckebusch, J.: Detection and localization of change points in temporal networks with the aid of stochastic block models. J. Stat. Mech.: Theory Exp. (2016)
28. Nobile, A.: Bayesian analysis of finite mixture distributions. Ph.D. Dissertation (1994)
29. Daudin, J.-J., Picard, F., Robin, S.: A mixture model for random graphs. Stat. Comput. **18**, 173–183 (2008)
30. Zanghi, H., Ambroise, C., Miele, V.: Fast online graph clustering via Erdös-Rényi mixture. Pattern Recognit. **41**, 3592–3599 (2008)
31. Nobile, A., Fearnside, A.T.: Bayesian finite mixtures with an unknown number of components: The allocation sampler. Stat. Comput. **17**, 147–162 (2007)
32. Luxburg, U.V.: A tutorial on spectral clustering. Stat. Comput. **17**, 395–416 (2007)
33. Wyse, J., Friel, N.: Block clustering with collapsed latent block models. Stat. Comput. **22**, 415–428 (2012)
34. Latouche, P., Birmele, E., Ambroise, C.: Variational Bayesian inference and complexity control for stochastic block models. Stat. Model. **12**, 93–115 (2012)