Inference for a generalised stochastic block model with unknown number of blocks and non-conjugate edge models

Matthew Ludkin[a]

[a]*Mathematics and Statistics, Lancaster University, Lancaster, United Kingdom, LA1 4YW.*

Abstract

The stochastic block model (SBM) is a popular model for capturing community structure and interaction within a network. Network data with non-Boolean edge weights is becoming commonplace; however, existing analysis methods convert such data to a binary representation to apply the SBM, leading to a loss of information. A generalisation of the SBM is considered, which allows edge weights to be modelled in their recorded state. An effective reversible jump Markov chain Monte Carlo sampler is proposed for estimating the parameters and the number of blocks for this generalised SBM. The methodology permits non-conjugate distributions for edge weights, which enable more flexible modelling than current methods as illustrated on synthetic data, a network of brain activity and an email communication network.

Keywords: network, stochastic block model, statistical analysis of network data, non-conjugate analysis

1. Introduction

Statistical analysis of networks has seen much growth in recent years with the increasing availability of network data. In this paper, a network consists of a set of nodes, which can form pairwise interactions. Each possible interaction is referred to as an *edge*, with the value of that interaction called an *edge weight*.

The aim of statistical network modelling is to describe the edge weights with a probabilistic model, potentially performing inference for model parameters. Such models include the exponential random graph (Snijders et al., 2006), the class of latent space models (Hoff et al., 2002) and the stochastic block model (SBM) (Frank and Harary, 1982; Holland et al., 1983). In the classic SBM, the set of nodes is partitioned into *blocks* such that the edge weight between two nodes depends on their block memberships. There is a rich literature on the SBM including both Bayesian and frequentist treatments. Extensions to the SBM include restricting the SBM to only within-block and between-block edge-weight distributions in the affiliation network (Snijders and Nowicki, 1997; Nowicki and Snijders, 2001; Copic et al., 2009), multiple-block memberships in the mixed-membership SBM (Airoldi et al., 2008), degree-corrected SBM (Karrer and Newman, 2011), and the infinite relational model (IRM), (Kemp et al., 2006) where the number of blocks is treated as unknown. For a thorough review of the SBM and inference methods, see Matias and Robin (2014).

Email address: m.ludkin1@lancaster.ac.uk (Matthew Ludkin)

This paper considers two extensions to the SBM: (i) modelling general edge weights (*i.e.* non-binary interaction data) and (ii) estimating the number of blocks. Previous authors have attempted extension (i) with a weighted or valued network (Jiang et al., 2009; Mariadassou et al., 2010; Ambroise and Matias, 2012) or considering a time-series of edge weights (Matias and Miele, 2017; Xin et al., 2017; Ludkin et al., 2018). Multiple methods have been considered for extension (ii); these fall into two main approaches: (a) a post-hoc analysis of multiple model fits using model selection techniques, and (b) treating the number of blocks as a random variable. Approach (a) includes likelihood-based methods using the Bayesian information criteria and its derivatives (Daudin et al., 2008; Latouche et al., 2012; Wang et al., 2017; Saldaña et al., 2017), information-based methods using minimum description lengths (Peixoto, 2013), sequential testing by embedding successive block models with an increasing number of blocks (Lei, 2016) and cross-validation (Chen and Lei, 2016). Approach (b) is achieved in a Bayesian framework by setting a prior for the number of blocks. Geng et al. (2019) use a mixture of finite mixtures representation, while the IRM (Mørup and Schmidt, 2013) uses a Chinese Restaurant Process (CRP) (Gershman and Blei, 2012).

Some authors (Mørup et al., 2011; Mørup and Schmidt, 2012, 2013; McDaid et al., 2013) have considered both extensions (i) and (ii) and posited collapsed Gibbs samplers to perform inference on the number of blocks, node membership and edge-weight model parameters. However, all of these methods require a conjugate model for the edge-weight distributions. This article aims to achieve both extensions by generalising the SBM to arbitrary edge-weight distributions and modelling the number of blocks in one Bayesian framework *without* the restriction of conjugate edge-weight distributions. This is highlighted in Section 5.2 where a negative binomial model is applied to the edge weights within an email network. Such a model cannot be applied using existing methodology since no conjugate prior distribution exists for the negative binomial with both parameters unknown. This approach greatly broadens the applicability of the general stochastic block model to network data with non-conjugate edge-weight distributions.

The proposed methodology to perform inference is a Markov chain Monte Carlo sampler which provides samples from the posterior distribution of the block parameters, block memberships and number of blocks. The sampling algorithm is inspired by Green and Richardson (2001) – a reversible jump Markov chain Monte Carlo (RJMCMC) (Green, 1995) scheme using split and merge proposals to explore the posterior by either combining two blocks, or splitting a block into two. Nobile and Fearnside (2007); McDaid et al. (2013) make use of a split-merge proposal, although due to the conjugate models considered, they do not require parameter values. The difficulty in designing an effective split-merge algorithm rests on ensuring that parameter values are "matched" when changing dimension. Previous authors have proposed sampling algorithms, such as the collapsed Gibbs sampler of McDaid et al. (2013) – for a given node, the posterior probability of belonging to a given block is computed with all other parameters fixed. Under the collapsed regime, assigning a node to a new block is simple, since the parameters have been integrated from the model. In the case of non-conjugate mixture models, the parameters are required to evaluate the likelihood of such a re-assignment; this added complexity can be handled within a full RJMCMC scheme as described in Section 3.

The remainder of the paper is organised as follows: in Section 2, the specifics of the generalised SBM are presented. Section 3 introduces the split-merge sampling algorithm. In Section 4, the sampler is applied to simulated data, whilst in Section 5, the split-merge sampler is used to analyse some real network data. Finally, closing remarks and extensions to the model and sampler are discussed in Section 6.

2. A generalisation of the stochastic block model

This section describes the stochastic block model and details the generalisation to arbitrary edge-weight distributions for network data.

Mathematically, a network is represented as a *weighted graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ where $\mathcal{V}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges and $\mathcal{W}$ is the set of edge weights. This paper uses the shorthand $ij \in \mathcal{E} \implies (i,j) \in \mathcal{E}$. The *weight* of edge ij is denoted by $W_{ij} \in \mathcal{W}$. To simplify exposition, it is assumed that all edge weights are observed, *i.e.* $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ and $W_{ij} \in \mathcal{W}$ for all $ij \in \mathcal{E}$. In this way, an un-weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be viewed as a weighted graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}', \mathcal{W}')$ with $\mathcal{E}' = \mathcal{V} \times \mathcal{V}$, $W'_{ij} = 1$ if $ij \in \mathcal{E}$ and $W_{ij} = 0$ otherwise. In the case where the network contains directed edges, the set $\mathcal{E}$ consists of ordered pairs such that $(i,j) \neq (j,i)$.

The canonical SBM (Holland et al., 1983; Fienberg et al., 1985; Wasserman and Anderson, 1987) can be viewed as such a weighted graph with $W_{ij} \in \{0, 1\}$, a fixed number of nodes ($|\mathcal{V}| = N$) and K blocks. The nodes are partitioned into blocks, with each node belonging to only one block. Let $\boldsymbol{Z}$ be the block indicator matrix with $Z_{ik} = 1$ if node i belongs to block k and 0 otherwise. As such, $\boldsymbol{Z}_i$ is a one-of-K indicator vector. It is assumed that $\boldsymbol{Z}_i$ is drawn from a multinomial distribution with parameter $\boldsymbol{\rho}$, a probability vector of length K which governs the block memberships. The prior probability that a node belongs to block k is given by ρ_k. Let $\boldsymbol{\theta}$ be a $K \times K$ matrix of edge-weight parameters, such that ϑ_{kl} is the probability that $W_{ij} = 1$ between nodes i and j in blocks k and l respectively. Note $\vartheta_{kl} = \boldsymbol{Z}_i^\top \boldsymbol{\vartheta} \boldsymbol{Z}_j$. This model is summarised in Equation (1); first the nodes are assigned to blocks, then – given these block memberships – the edge weights are drawn with parameters depending on the block membership of the end nodes.

$$\begin{aligned} \boldsymbol{Z}_i | \boldsymbol{\rho} &\overset{\text{iid}}{\sim} \text{Multinomial}(\boldsymbol{\rho}) \,, \\ W_{ij} | \boldsymbol{\vartheta}, \boldsymbol{Z} &\overset{\text{ind}}{\sim} \text{Bernoulli}(\boldsymbol{Z}'_i \boldsymbol{\vartheta} \boldsymbol{Z}_j) \,. \end{aligned} \tag{1}$$

In full generality, there are $K(K+1)/2$ free parameters in $\boldsymbol{\vartheta}$ for an un-directed network (or K^2 for a directed network). In the affiliation model (Snijders and Nowicki, 1997; Nowicki and Snijders, 2001; Copic et al., 2009), $\boldsymbol{\vartheta}$ is restricted to two parameters, one each for between-block ($\vartheta_{kl}, k \neq l$) and within-block (ϑ_{kk}) interactions.

In this article, a parameterisation between these two extremes is considered: let θ_k be the parameters governing edge weights between nodes belonging to block k, and a global parameter θ_0 for edge weights between nodes in different blocks. In this way, the number of parameters is $K + 1$, and grows linearly in the number of blocks. This model is appropriate for networks where between-block connections are relatively homogeneous; for example, in ecological contact networks, where herds of animals remain close together for most of the time, with some interactions between herds. Let $\boldsymbol{\theta}$ be the matrix of parameters with $\theta_{kk} = \theta_k$ and $\theta_{kl} = \theta_0$ for $k = 1, \ldots, K, l \neq k$, then the quadratic form $\boldsymbol{Z}_i^\top \boldsymbol{\theta} \boldsymbol{Z}_j$ picks the parameter governing the edge weight W_{ij}.

With this parameterisation, the classic SBM in Equation (1) is extended to allow the number of blocks to be random and to model general edge weights, such as count or continuous data. Let G and G_0 be the distribution on the edges-weights and parameters respectively. Prior parameters $\boldsymbol{\alpha}$ are assigned to the block parameters $\boldsymbol{\theta}$. Since the number of blocks K is considered unknown, a prior must be placed on both the number of blocks and block memberships. Let F be a joint distribution for $(K, \boldsymbol{Z})$ with parameters γ and δ then the generalised form of the SBM considered in this paper is:

$$
\begin{aligned}
K, \boldsymbol{Z} &\sim \mathrm{F}(\gamma, \delta)\,, \\
\theta_k &\overset{\text{ind}}{\sim} \mathrm{G}_0(\boldsymbol{\alpha})\,, \\
W_{ij}|\boldsymbol{\theta}, \boldsymbol{Z} &\overset{\text{ind}}{\sim} \mathrm{G}(\boldsymbol{Z}_i'\boldsymbol{\theta}\boldsymbol{Z}_j)\,.
\end{aligned} \tag{2}
$$

This framework may be extended to an edge-weight distribution G with multiple parameters. For example, if G represents the normal distribution, then $\theta_k = (\mu_k, \sigma_k)$ represents the mean and standard deviation of the edge weights in block k. In this case, an additional subscript is required on θ_k such that θ_{kp} is the p^{th} parameter for block k. In the normal example, line 3 of Equation (2) yields $W_{ij}|\boldsymbol{\theta}, \boldsymbol{Z} \overset{\text{ind}}{\sim} \text{Normal}(\boldsymbol{Z}_i'\boldsymbol{\mu}\boldsymbol{Z}_j, \boldsymbol{Z}_i'\boldsymbol{\sigma}\boldsymbol{Z}_j)$.

The choice of distributions for G and G_0 is driven by the type of edge weight considered (*i.e.* edge weights representing counts could be modelled using a Poisson distribution for G). On the other hand, there is flexibility for distribution F. As discussed in Geng et al. (2019), the popular choice of the Chinese Restaurant Process (CRP) yields the undesirable property that large probability is assigned to blocks with relatively few nodes. Indeed, Miller and Harrison (2018) show that using a CRP prior on $(K, \boldsymbol{Z})$ in mixture models leads to inconsistent estimation of the number of clusters, even in the asymptotic regime when N tends to infinity. To circumvent this, Miller and Harrison (2018) propose using the "mixture of finite mixtures approach" (MFM) where the number of blocks has an explicit prior distribution. Let F_0 be a distribution on $\{1, 2, 3, \ldots\}$ with parameter δ, then the prior for $(K, \boldsymbol{Z})$ considered in the remainder of the paper is given in Equation (3):

$$
\begin{aligned}
K &\sim F_0(\delta), \\
\rho|K &\overset{\text{ind}}{\sim} \text{Dirichlet}(\gamma, K)\,, \\
\boldsymbol{Z}_i|\boldsymbol{\rho} &\overset{\text{ind}}{\sim} \text{Multinomial}(\boldsymbol{\rho})\,,
\end{aligned} \tag{3}
$$

where $\text{Dirichlet}(\gamma, K)$ is the symmetric Dirichlet distribution on the $K-1$ simplex. The *size* of block k is the number of nodes whose block membership is k and is given by $N_k = \sum_{i=1}^{N} Z_{ik}$. Let $\boldsymbol{N} = \{N_k : k = 1, \ldots, K\}$ be the set of block sizes, then the distribution for $\boldsymbol{N}$ under the CRP and the MFM are:

$$
p_{\text{CRP}}(\boldsymbol{N}) = \prod_{k=1}^{K} N_k^{-1} \quad \text{vs.} \quad p_{\text{MFM}}(\boldsymbol{N}) = \prod_{k=1}^{K} N_k^{\gamma-1}.
$$

Notice that the MFM gives comparatively less probability mass to small blocks than the CRP. Also, the distribution for the CRP is independent of γ. Thus, the MFM approach gives more control over the prior block structure.

The parameter $\boldsymbol{\rho}$ can be marginalised out of Equation (3) to obtain a prior density for block memberships depending only on K and γ as such:

$$
\mathrm{f}(\boldsymbol{Z}|\gamma, K) = \int_{\boldsymbol{\rho}} \mathrm{f}(\boldsymbol{Z}|\boldsymbol{\rho})\, \pi_0(\boldsymbol{\rho}|\gamma)\, d\boldsymbol{\rho} = \int_{\boldsymbol{\rho}} \prod_{k=1}^{K} \rho_k^{N_k+\gamma+1} \frac{\Gamma(K\gamma)}{\Gamma(\gamma)^K} d\boldsymbol{\rho} = \frac{\Gamma(K\gamma)}{\Gamma(\gamma)^K} \frac{\prod_{k=1}^{K} \Gamma(\gamma + N_k)}{\Gamma(K\gamma + N)},
$$

since $\sum_{k=1}^{K} N_k = N$ and where $\Gamma(a) = \int_0^\infty x^{a-1} e^x dx$ is the gamma function; this is referred to as the

Dirichlet-Multinomial distribution. Similarly, the conditional distribution for the block membership of node i, given K and the other block memberships $\boldsymbol{Z}_{-i}$ is:

$$\begin{aligned}\mathrm{f}(\boldsymbol{Z}_i|\boldsymbol{Z}_{-i},K,\gamma) &= \frac{\mathrm{f}(\boldsymbol{Z}|\gamma,K)}{\mathrm{f}(\boldsymbol{Z}_{-i}|\gamma,K)} = \frac{\prod_{k=1}^{K}\Gamma(\gamma+N_k)}{\Gamma(K\gamma+N)}\frac{\Gamma\left(K\gamma+N-\sum_{k=1}^{K}Z_{ik}\right)}{\prod_{k=1}^{K}\Gamma(\gamma+N_k-Z_{ik})} \\ &= \frac{1}{K\gamma+N-1}\prod_{k=1}^{K}\frac{\Gamma(\gamma+N_k)}{\Gamma(\gamma+N_k-Z_{ik})},\end{aligned}$$

since $\sum_{k=1}^{K} Z_{ik} = 1$ and $x\,\Gamma(x) = \Gamma(x+1)$. Therefore,

$$\mathrm{f}(Z_{il}=1|\boldsymbol{Z}_{-i},K,\gamma) = \frac{\gamma+N_l-1}{K\gamma+N-1}.$$

In the remainder of this article, the generalised SBM (GSBM) used is:

$$\begin{aligned} K-1 &\sim \mathrm{Pois}(\delta)\,, \\ \boldsymbol{Z}|K &\overset{\text{ind}}{\sim} \text{Dirichlet-Multinomial}(\gamma,K)\,, \\ \boldsymbol{\theta}_k &\overset{\text{ind}}{\sim} \mathrm{G}_0(\boldsymbol{\alpha})\,, \\ W_{ij}|\boldsymbol{\theta},\boldsymbol{Z} &\overset{\text{ind}}{\sim} \mathrm{G}(\boldsymbol{Z}_i'\boldsymbol{\theta}\boldsymbol{Z}_j)\,, \end{aligned} \tag{4}$$

where G_0 and G are specified by the modeller. The prior on $(K,\boldsymbol{Z})$ will be referred to as the DMA(γ,δ) (Dirichlet-Multinomial Allocation) prior. When a model G is defined, we refer to the specific form of the model as G-SBM.

3. Split-merge sampler

This section discusses the benefit of split-merge steps over Gibbs samplers for mixture models, describes the difficulty that arises when designing split-merge moves for block membership in the GSBM, and presents a split-merge RJMCMC sampler for the GSBM. This algorithm draws samples from the posterior distribution of $(K,\boldsymbol{Z},\boldsymbol{\theta})$.

For models containing a mixture component (such as the block structure in Mørup and Schmidt, 2012; McDaid et al., 2013) a Gibbs sampler can get stuck in local modes of the posterior. Consider two "true" blocks k and l with sizes $N_k \geq N_l$ and a state s of a Gibbs sampler with a block k^s consisting of all nodes in true blocks k and l. For the Gibbs sampler to separate the nodes in k^s into blocks k and l, it will require at least N_l steps, each of which takes a node assigned to k^s and assigns it to a new block l^s. Each of these moves is quite unlikely, especially if the parameters $\boldsymbol{\theta}_k,\boldsymbol{\theta}_l$ are close to $\boldsymbol{\theta}_0$. On the other hand, if all nodes could be moved at once, then the proposal would be more likely to be accepted. This is a common problem with Gibbs sampling algorithms: the one-at-a-time nature of the algorithm means large changes in posterior space are unlikely, even if the combined changes increase the posterior considerably. One way to address this is to use a split-merge sampler.

Split-merge samplers have been developed for general mixture models (Green and Richardson, 2001), with emphasis on a mixture of normal densities. In a standard parametric mixture model, each component has a different form (either different distributions or different parameter values) and each data point is drawn from a component of the mixture. A split-merge sampler applied to such a data set explores the possible assignments of data points to components by successively proposing to either merge two components together or split one component in two. Care must be taken when designing such proposals: they must be an isomorphism and differentiable to ensure the validity of the underlying Markov chain. Furthermore, to be efficient, a proposed structure should have similar posterior support to the current structure to give a reasonable probability of acceptance. Notice that, since each data point belongs to one component, a split move which assigns a data point to a new cluster will be penalised by the prior on the number of components, but the likelihood will increase if the parameter for the new component is a good fit for the assigned data point. Compare this to the latent block membership of the GSBM: reassigning a node i to a new block *affects all nodes with an edge to* i. This implies that the prior will penalise the split move for adding a block for the new node, and the likelihood will penalise based on the $(N-1)$ edge weights incident to i. Therefore, when considering split-merge samplers for the GSBM, multiple edge weights are affected by changing the block membership of one node; this fact complicates the design of a successful proposal.

The remainder of this section introduces the split-merge sampler for the GSBM. The sampler consists of four moves: re-sampling parameter values, splitting or merging blocks, reassigning nodes to the current set of blocks, and adding or deleting an empty block.

Let $(K^s, \boldsymbol{Z}^s, \boldsymbol{\theta}^s)$ be the value of the parameters in step s of the sampler. Values for parameter $\boldsymbol{\theta}$ given the block structure can be sampled using any MCMC kernel. In this work, each θ_i is re-sampled using a random walk on a transformed scale. The difficult proposals are trans-dimensional: merging and splitting blocks. These are described in the following subsections. The full split-merge algorithm is given in Algorithm 1.

Merge move

The merge proposal takes a state $(K^s, \boldsymbol{Z}^s, \boldsymbol{\theta}^s)$ and proposes a new state $(K', \boldsymbol{Z}', \boldsymbol{\theta}')$. Such a move will reduce the number of blocks by one: $K' = K^s - 1$. Firstly, two blocks k and l are sampled to merge – possible mechanisms include choosing blocks proportional to block size, inversely proportional to block size, at random, etc. In this paper, for simplicity, the pair k, l is chosen with probability $1/K^s(K^s-1)$. Secondly, the block membership $\boldsymbol{Z}'$ is updated. This is deterministic: any node that is a member of block k or l in $\boldsymbol{Z}^s$ is assigned to block k' in $\boldsymbol{Z}'$. All other nodes keep their block assignment. Next, the parameter values are updated. Following the recommendations of Green and Richardson (2001), proposing a value $\boldsymbol{\theta}'_{k'}$ with similar explanatory power as $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_l$ should ensure that $\boldsymbol{\theta}'_{k'}$ is well supported in the posterior. A simple approach is to take the mean value: $\boldsymbol{\theta}'_{k'} = \boldsymbol{\theta}_k/2 + \boldsymbol{\theta}_l/2$; however, to allow more flexibility in the sampler, an uneven merge is considered using a weighted mean with tuning parameter $\lambda \in (0, 1)$. Since the split move will invert the merge move, a *matching function* m is required to ensure that parameters lie in the correct space. For example, a rate parameter must be positive, whereby a suitable choice for m is the exponential function. Possible matching functions for some common parameter spaces are shown in Table 1. The full parameter proposal during a merge move is shown in Equation (5):

$$\mathrm{m}(\boldsymbol{\theta}'_{k'}) = \lambda\,\mathrm{m}(\boldsymbol{\theta}_k) + (1-\lambda)\,\mathrm{m}(\boldsymbol{\theta}_l) \tag{5}$$

Algorithm 1 Reversible jump Markov Chain Monte Carlo sampler for the GSBM with unknown K: split-merge algorithm.

Inputs: edge-weight data $\boldsymbol{w}$, prior parameters $\boldsymbol{\alpha}, \gamma, \delta$, sampler parameters λ, ν, σ.
Draw $K^0, \boldsymbol{Z}^0 \sim \mathrm{F}_0(\cdot|\gamma, \delta)$.
Draw $\boldsymbol{\theta}^0 \sim \mathrm{G}_0(\cdot|\boldsymbol{\alpha})$.
for $s = 1, \ldots, S$ **do**
  Draw $\boldsymbol{\theta}^s \sim \mathrm{Update}\big(\cdot|\boldsymbol{w}, K^{s-1}, \boldsymbol{Z}^{s-1}, \boldsymbol{\theta}^{s-1}, \boldsymbol{\alpha}\big)$
  Let $K^s = K^{s-1}$
  if K^s=1 **then**
    Propose a split
  else
    with probability 1/2 propose a split or a merge
  end if
  if There are no empty blocks **then**
    Propose adding an empty block
  else
    with probability $\frac{N_\emptyset}{N_\emptyset+\nu}$ attempt deleting an empty block.
    or with probability $\frac{\nu}{N_\emptyset+\nu}$ attempt adding an empty block.
  end if
  for $i = 1, \ldots, N$ **do**
    for $k = 1, \ldots, K^s$ **do**
      Let $p_k = \mathrm{g}(w_{i\cdot}|\boldsymbol{Z}_{-i}, Z_{ik} = 1, \boldsymbol{\theta})\,\mathrm{f}(Z_{ik} = 1|\boldsymbol{Z}_{-i})$
    end for
    Draw $\boldsymbol{Z}'_i \sim \mathrm{Multinomial}(\boldsymbol{p})$
  end for
  Store sample $(\boldsymbol{Z}^s, \boldsymbol{\theta}^s, K^s)$.
end for
return samples $\boldsymbol{Z}, \boldsymbol{\theta}, K$

Finally, the acceptance probability A_{merge} is computed (see Appendix A) and the next state of the sampler $(K^{s+1}, \boldsymbol{Z}^{s+1}, \boldsymbol{\theta}^{s+1})$ is taken as $(K', \boldsymbol{Z}', \boldsymbol{\theta}')$ with probability A_{merge}, and as $(K^s, \boldsymbol{Z}^s, \boldsymbol{\theta}^s)$ otherwise.

Table 1: Possible matching functions to ensure parameters lie in the correct space.

Range for $\boldsymbol{\theta}$	Possible matching function m
(∞, ∞)	$\mathrm{m}(x) = x$
$[0, \infty)$	$\mathrm{m}(x) = \log(x)$
$[0, 1]$	$\mathrm{m}(x) = \mathrm{logit}(x) = \log(x) - \log(1-x)$

Split move

The split proposal takes a state $(K^s, \boldsymbol{Z}^s, \boldsymbol{\theta}^s)$ and proposes a new state $(K', \boldsymbol{Z}', \boldsymbol{\theta}')$ with $K' = K^s + 1$. Firstly, the block to split is chosen at random. Possible mechanisms include sampling at random among the K^s blocks, proportional to block size, etc. In this paper the block is chosen uniformly amongst the K^s blocks. To mirror the notation of the merge move, the block to split is labelled k', and the proposed new blocks k and l.

The first step in a split move determines the new block parameters. This requires the inverse of Equation (5). On top of this, an auxiliary variable u' is needed to match the dimension of the parameter space. In this work, $u' \sim \mathrm{Normal}\left(0, \sigma^2\right)$ and represents the weighted difference of the mapped parameters $\mathrm{m}(\theta_k)$ and $\mathrm{m}(\theta_l)$. The parameter split is thus:

$$\mathrm{m}(\boldsymbol{\theta}_k) = \frac{\mathrm{m}(\boldsymbol{\theta}'_{k'}) + u'}{2\lambda'} \qquad \mathrm{m}(\boldsymbol{\theta}_l) = \frac{\mathrm{m}(\boldsymbol{\theta}'_{k'}) - u'}{2(1-\lambda')}$$

Note that the dimension-matching criterion of RJMCMC (Green, 1995) is achieved since the vectors $(\boldsymbol{\theta}'_{k'}, u', \lambda')$ and $(\boldsymbol{\theta}_k, \boldsymbol{\theta}_l, \lambda)$ have the same cardinality.

To determine $\boldsymbol{Z}'$, the nodes assigned to block k' in $\boldsymbol{Z}^s$ are reassigned to blocks k and l. In a similar fashion to Green and Richardson (2001), nodes are assigned sequentially to either block k or l proportional to the model likelihood. It is not possible to compute the full likelihood during this procedure for the GSBM because edge weights exist between all nodes. Specifically, let i and j be the only nodes in block k'. Choosing to assign i to block k or l proportional to the likelihood requires knowledge of the block membership of j, which does not yet exist. The quantity can be calculated in principle by looking at all the possible allocations of the nodes in block k to k' and l'. This operation is expensive; instead, it is estimated by the following sequential process:

First, all nodes in block k' are unassigned and placed in a holding set $\mathcal{I}$. The set of remaining nodes is labelled $\mathcal{J}$ and the current set of block assignments $\boldsymbol{Z}_{\mathcal{J}}$. Take a permutation $\sigma(\mathcal{I})$ of $\mathcal{I}$ – this is the order in which nodes will be reassigned to block k or l.

When assigning node i, the following quantity can be calculated:

$$\mathrm{q}(Z'_i = k') = \frac{\mathrm{f}\left(\boldsymbol{w} | Z'_i = k', \boldsymbol{Z}'_{\mathcal{J}}, \boldsymbol{\theta}'\right)}{\mathrm{f}\left(\boldsymbol{w} | Z'_i = k', \boldsymbol{Z}'_{\mathcal{J}}, \boldsymbol{\theta}'\right) + \mathrm{f}\left(\boldsymbol{w} | Z'_i = l', \boldsymbol{Z}'_{\mathcal{J}}, \boldsymbol{\theta}'\right)}.$$

Node i is then assigned to block k with probability $\mathrm{q}(Z'_i = k)$ and to block l otherwise. Once assigned, i is moved from $\mathcal{I}$ to $\mathcal{J}$ for the next assignment.

The total proposal probability of the new block assignment is thus:

$$\mathrm{q}(\boldsymbol{Z}') = \prod_{i \in \sigma(\mathcal{I})} \mathrm{q}(Z'_i = k)^{\mathbb{I}[Z'_i = k']} \left(1 - \mathrm{q}(Z'_i = k)\right)^{\mathbb{I}[Z'_i = l']}.$$

Finally, the proposed split is accepted as the next state of the sampler with probability A_{split} as in Equation (A.1), Appendix A.

Gibbs reassignment move

To allow the sampler to explore the parameter space, an additional two moves are included: a Gibbs-like move (which allocates each node to a block proportional to the posterior density) and a move that allows the addition and deletion of empty blocks.

The Gibbs-like allocation move for node i computes the conditional posterior value for i being a member of each of the K blocks in the current state of the sampler. Since K is finite, this set of posterior values can trivially be normalised to a probability vector, such that p_{ik} is the probability that node i is reassigned to block k. Thanks to the structure of the GSBM, p_{ik} can be written as the product of two densities: the posterior density of edge weights to nodes in block k, and the posterior density of edge weights to nodes in other blocks:

$$\begin{aligned} p_{ik} &= \mathrm{p}(Z_{ik} = 1 | \boldsymbol{Z}_{-i}, \boldsymbol{w}, \boldsymbol{\theta}), \\ &\propto \mathrm{f}(\boldsymbol{Z}_{ik} = 1 | \boldsymbol{Z}_{-i}) \prod_{j \neq i} \mathrm{g}(w_{ij} | \boldsymbol{Z}_j, bZ_{ik} = 1, \boldsymbol{\theta}), \\ &= \mathrm{f}(\boldsymbol{Z}_{ik} = 1 | \boldsymbol{Z}_{-i}) \prod_{j \neq i} \mathrm{g}(w_{ij} | \boldsymbol{\theta}_k)^{\boldsymbol{Z}_{jk}} \, \mathrm{g}(w_{ij} | \boldsymbol{\theta}_0)^{1 - \boldsymbol{Z}_{jk}}. \end{aligned}$$

Notice it is possible to reassign i to its current block. This move, as well as the split move, can leave a block empty; waiting for the sampler to merge an empty block with another block can leave empty blocks in the sampler state for some time, adding to the uncertainty around the number of blocks K. A proposal that addresses these concerns is considered in the next section.

Add or delete empty blocks

The second extension allows for the deletion and addition of empty blocks; the *delete empty block* move is the inverse of *add empty block*. During the *delete empty block* move, a candidate block is chosen at random from the current set of empty blocks. When an empty block is added, it is given the label $K + 1$. For simplicity, when an add/delete move is attempted, the probability of adding a block is chosen proportional to a sampler parameter ν. The probability of choosing to delete an empty block is proportional to the number of empty blocks in the current state, $N_\emptyset$. Note that the likelihood of the edge weights does not change with the addition of empty blocks since the entire node structure remains unaffected. When a block is added, a parameter $\boldsymbol{\theta}^*$ is drawn from the prior distribution G_0. The acceptance probabilities of the add and delete empty block moves are calculated as:

$$A_{add} = \frac{\pi_0(K+1, \boldsymbol{Z})}{\pi_0(K, \boldsymbol{Z})} \frac{\nu + N_\emptyset}{\nu(\nu + N_\emptyset + 1)}, \quad \text{and} \quad A_{del} = \frac{\pi_0(K-1, \boldsymbol{Z})}{\pi_0(K, \boldsymbol{Z})} \frac{\nu(\nu + N_\emptyset)}{\nu + N_\emptyset - 1}.$$

The sampler is implemented in the `R` package "SBMSplitMerge" Ludkin (2020). This package is used to perform the inference in the following sections.

4. Simulated data

In this section, the split-merge sampler of Section 3 is demonstrated on simulated data. The scripts to generate these example networks, run the sampler, and produce the figures (as well as the data in Section 5) are available on GitHub (`https://github.com/ludkinm/SBMSplitMerge/releases/tag/CRAN-1.1.1`).

Two data sets are considered. Both consist of 100 nodes split into four blocks with sizes 19, 23, 27 and 31. Each network has the same block structure. The first data set uses a Bernoulli distribution as its edge-weight distribution G. The second data set uses a generalised negative binomial distribution. Data was simulated from the edge-weight distributions with and plotted in Figure 1a for the Bernoulli data set, then Figure 2a for the negative binomial.

The generalised negative binomial distribution is parameterised by the real-valued "number of failures" $r > 0$ and success probability $p \in [0, 1]$. If $X \sim \text{NegBin}(r, p)$ then:

$$\mathbb{P}(X = x) = \frac{\Gamma(x + r)}{\Gamma(r)\, x!} p^r (1 - p)^x, \text{ for } x = 0, 1, 2, \ldots$$

Notice that the Bernoulli distribution admits a conjugate prior; therefore, existing samplers, such as those introduced by Mørup and Schmidt (2012) and McDaid et al. (2013), could be applied. However, for the negative binomial with both r and p unknown, no conjugate prior exists.

To apply the GSBM, the prior on K and $\boldsymbol{Z}$ was set to a DMA distribution with hyperparameters set to $(\gamma, \delta) = (1, 10)$. The parameter values used for each of the edge-weight models is given in Table 2. For the network with Bernoulli-distributed edge weights, the uniform prior Beta(1, 1) was applied to each parameter $\boldsymbol{\theta}$. In the negative binomial network with both parameters unknown, a Beta(1, 1) distribution is placed on the probability parameter p and the prior for r is set to Gamma(1, 1).

Table 2: Simulated data parameter values for each edge-weight distribution.

Parameter	θ_0	θ_1	θ_2	θ_3	θ_4
Bernoulli(p)	0.05	0.4	0.5	0.6	0.7
Negative binomial(p, r)	(0.5, 1)	(0.5, 1)	(0.5, 4)	(0.5, 5)	(0.5, 6)

In both cases, a random walk Metropolis-Hastings step was applied to $\boldsymbol{\theta}$ on a transformed scale with standard-deviation 0.1. A draw from the prior was taken as the initial state then the split-merge sampler of Section 3 ran for 10,000 iterations with 5000 iterations discarded as burn-in.

To evaluate the performance of the algorithm, the ability to detect the true number of blocks, block structure and parameter values are considered. To measure the ability to detect block structure, the posterior joint probabilities that two nodes belong to the same block are calculated after burn-in, via:

$$P_{ij} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathbb{I}[Z_{is} = Z_{js}], \tag{6}$$

where $\mathcal{S}$ contains the indices of samples remaining after burn-in.

The parameter estimates can be compared to the true values in Table 2. Note that the model in Equation (4) is invariant to a permutation of the block labels; this implies that the true and inferred structure may be the same up to a permutation of the block labels. To correct for this phenomenon, a permutation of the modal block labels under the MCMC to the true labels is derived and applied

to the parameters and block labels in the Markov chain (Details are given in Appendix B). Note this matching is only required to compare the true parameter values to the MCMC output.

The posterior joint probability that two nodes are in the same block (after burn-in) is displayed for the Bernoulli network in Figure 1b. This matches the truth very well: nodes who truly are in the same block have high posterior probability of being assigned to the same block (Equation 6), and nodes who are not in the same block have low posterior probability. The trace plot for K shows that for most iterations the sampler had four blocks, matching the truth, but explored some states with five or six blocks. The posterior modes of the parameters, and the 5% and 95% posterior confidence intervals are shown in Table 3. The posterior modes are all close to the true values in Table 2 for the Bernoulli network.

For the negative binomial network, Figure 2b shows that blocks 2, 3 and 4 are well identified by the sampler. As for the block 1, recall $\theta_0 = \theta_1$ in the true parameters; this gives no structure to block 1. Indeed, one could reassign the nodes in block 1 arbitrarily between two blocks 1a and 1b with $\theta_{1a} = \theta_{1b} = \theta_1$ and the likelihood would be unchanged. (Note this is not true for block $k = 2, 3, 4$ since some within-block interactions governed by $\theta_k \gg \theta_0$ would be governed by θ_0 under such a reassignment.) The sampler is able to explore regions of the posterior where nodes in block 1 are separate from the other nodes, as seen by the low probability region in the off-diagonal in Figure 2b. There is uncertainty around if the nodes in block 1 are in the same block as indicated by the range of posterior probabilities in the lower left block of Figure 2b. The estimated parameter values in Table 3 lead to similar conclusions: the estimates for parameters $\theta_0, \theta_2, \theta_3$ and θ_4 are good, but, the poor specification of block 1 leads to poor estimates of θ_1.

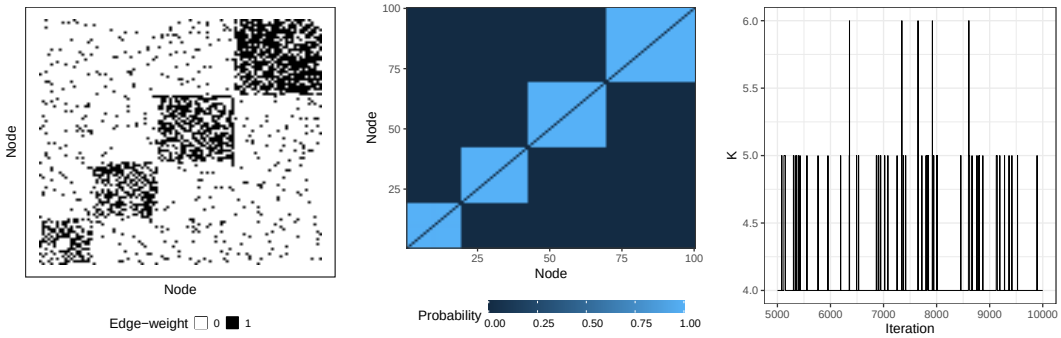
Table 3: Mode, 5% and 95% posterior quantiles for parameters in example networks.

Model	Bernoulli	Negative Binomial	Negative Binomial
Parameter	p	p	r
θ_0	0.052 (0.046, 0.058)	0.472 (0.442, 0.497)	0.895 (0.801, 0.978)
θ_1	0.425 (0.366, 0.491)	0.436 (0.059, 0.997)	0.642 (0.001, 1.575)
θ_2	0.506 (0.453, 0.557)	0.467 (0.392, 0.536)	3.196 (2.410, 4.126)
θ_3	0.638 (0.598, 0.677)	0.536 (0.472, 0.600)	5.545 (4.330, 7.183)
θ_4	0.678 (0.643, 0.714)	0.477 (0.425, 0.532)	5.392 (4.480, 6.692)

Assessing the convergence of a reversible jump Markov chain is non-trivial. Two techniques are applied in this section: (i) applying the Gelman-Rubin convergence statistic (Gelman and Rubin, 1992) to a summary statistic and (ii) starting two independent samplers from extreme block configurations – one with all nodes assigned to one block and the other with each node assigned to a unique block.

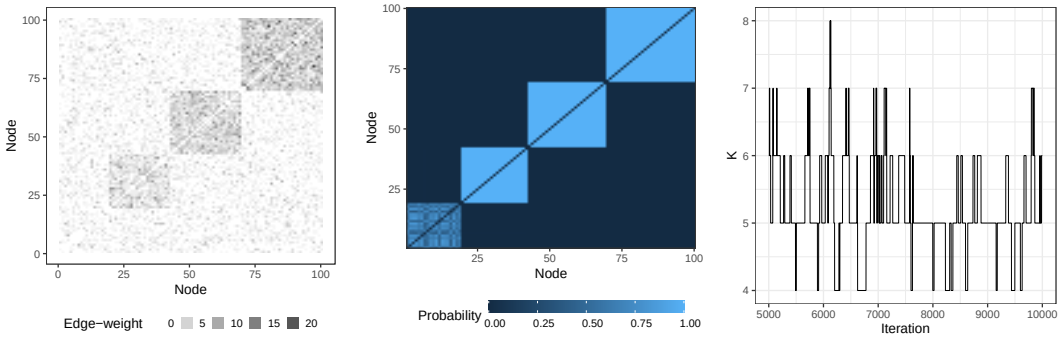
In the first case, the mean and variance of the parameter values are used as summary statistics of the sampler performance, which are recorded at every iteration of the sampler. The Gelman-Rubin statistics for the sampler for each model are shown in Table 4 based on 30 independent chains. These values are close to 1, indicating that convergence appears to have occurred during the first 10,000 iterations.

The second technique for assessing convergence is inspired by perfect simulation: starting two samplers at opposite extremes of the parameter space and observing both converging to the same area of the posterior indicates that the underlying Markov chains have converged. This process was used for the simulated data sets; trace plots for the number of blocks in each case are shown in Figure 3.



(a) Edge-weight matrix W (b) Joint posterior probability of belonging to same block P (c) Trace plot for K

Figure 1: Bernoulli edge weights: adjacency matrix and posterior summaries for block membership and number of blocks K.



(a) Edge-weight matrix W (b) Joint posterior probability of belonging to same block P (c) Trace plot for K

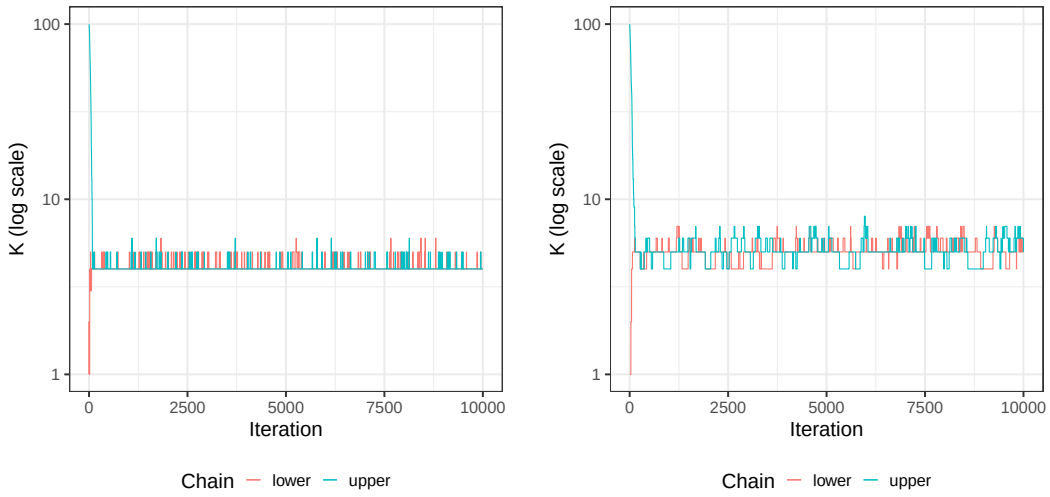
Figure 2: Negative binomial edge weights: adjacency matrix and posterior summaries for block membership and number of blocks K.

Model	Bernoulli	Negative binomial
Mean	1.0005 (1.0007)	1.0098 (1.0153)
Variance	1.0005 (1.0006)	1.0069 (1.0106)

Table 4: Rubin-Gelman statistics (and upper bound of 95% confidence interval) for each model with 30 independent chains of 10000 iterations.

5. Real data

The split-merge sampler is demonstrated on real networks: a network of brain connectivity with binary edge weights in Section 5.1 and a network of emails with count data for edge weights in Section 5.2.



(a) Bernoulli: Perfect simulation trace plot for K

(b) Negative binomial: Perfect simulation trace plot for K

Figure 3: Trace plots for number of blocks K in example networks. Two chains are simulated in each case: the "lower chain" with all nodes initially in one block (orange line) and the "upper chain" with all nodes initially assigned to different blocks (teal line).

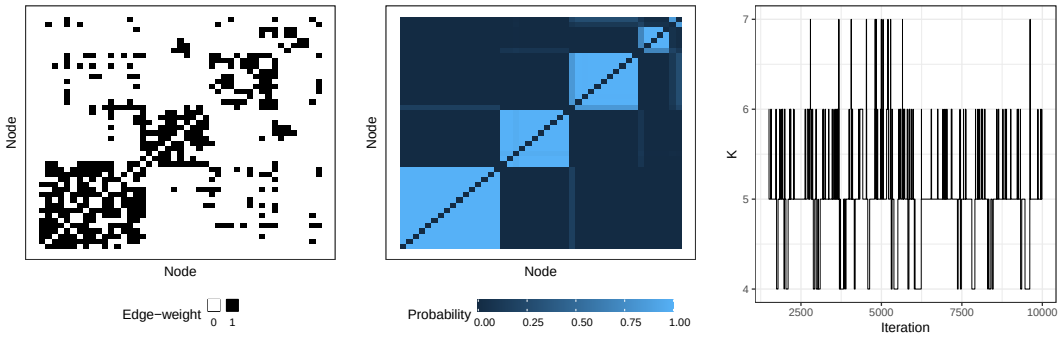
5.1. Macaque sensory data

The first data set analysed concerns the brain of a macaque monkey (Négyessy et al., 2006). Regions of the cortex were deemed connected, or not, during a sensory task. In total, 45 regions of the brain were analysed as a network.

A block model was proposed to partition the regions of the brain. This model assigns regions of the brain to the same block if their neural activity is similar. Since the data only provides binary edge weights, a Bernoulli-SBM is applied. A Beta(1,1) prior was placed on the edge probability parameters θ_k and a DMA(1,6) prior is placed on $(K, \boldsymbol{Z})$ for the block structure, thus the prior expected number of blocks is five. The split-merge algorithm was run for 10,000 iterations to provide samples from the posterior distribution of both block membership and parameter values. 1500 samples were discarded as burn-in.

Figure 4 displays posterior summaries for the split-merge sampler. A trace plot for the number of blocks, K, is shown in Figure 4c. This shows that the sampler settles on between four and six blocks with mode five. The joint posterior probability matrix P was calculated using Equation (6) and the modal block assignments were calculated from the MCMC chain output. Using the modal assignments, the nodes are ordered by block label. This ordering applied to the edge-weight matrix W and P are shown in Figure 4a and 4b respectively. The five blocks can be seen in Figure 4b as shown by the light blue regions. Counting from the lower left of Figure 4b, block five consists of two nodes; these nodes also have some probability of belonging to block three, as indicated by the shading in the final two columns/rows. Similarly, some uncertainty is displayed in the block membership of the first nodes in blocks three and four. Modal parameter estimates are shown in Table 5 together with 5% and 95% quantiles and the effective sample size. The parameters for smaller blocks have wider confidence intervals; this is expected since there are fewer edge weights

governed by those parameters. Note that parameter θ_5 is more uncertain; this is due to the block consisting of two nodes, meaning that θ_5 only governs one edge weight. The effective sample size cannot be computed for this parameter since it is absent in many iterations when the block has been merged with another block.



(a) Edge-weight matrix W (b) Joint posterior probability of belonging to same block P (c) Trace plot for K

Figure 4: Posterior summaries for block membership in macaque brain network ordered by modal block assignments.

Table 5: Modal parameter estimates, 95% posterior quantiles and effective sample sizes for macaque network.

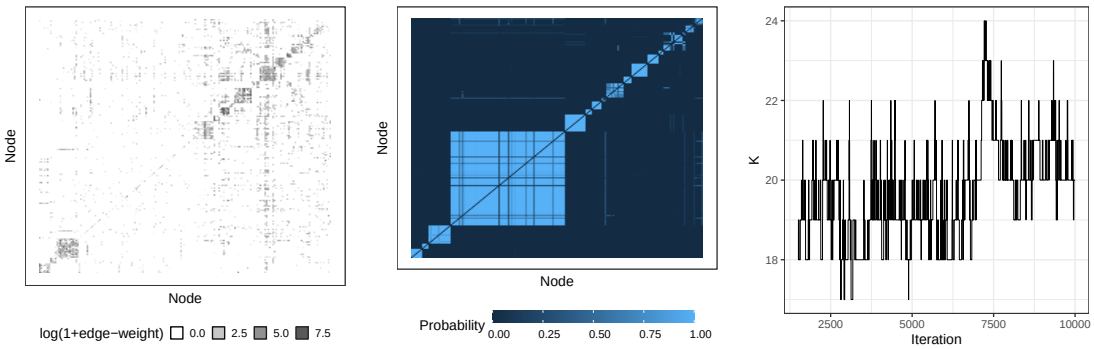
Parameter	Mode	5%	95%	Effective sample size
θ_0	0.09	0.08	0.11	1048
θ_1	0.70	0.64	0.75	553
θ_2	0.72	0.63	0.80	251
θ_3	0.56	0.43	0.68	126
θ_4	0.58	0.36	0.82	71
θ_5	0.70	0.15	0.99	NA

5.2. Enron emails

The Enron corporation was declared bankrupt in 2001 and later multiple employees were found guilty of accounting fraud. As a result of the trial, a corpus of emails leading up to the closure of the company was released as a public data set (Klimt and Yang, 2004). Aggregate counts of emails between any two employees are arranged into an edge-weight matrix. Note that this network contains directed edges and self-loops (since some emails are sent to mailing lists, to which the sender belongs). Two models for the edge weights were considered for this model: (i) a Poisson with a Gamma(1,1) prior and (ii) a negative binomial with a Gamma(1,1) prior for r and a Beta(1,1) prior for p. In both cases a DMA(1,10) joint prior is placed on $K, \boldsymbol{Z}$. On a first analysis, the mean number of emails sent by any one employee is 3.7, whilst the variance is 4753, so a Poisson model seems a bad fit *a priori*. The split-merge algorithm of Section 3 was applied with 10,000 iterations and 1500 discarded as burn-in.

As in Section 5.1, the joint posterior probability matrix P was calculated using Equation (6) and the modal block assignments were calculated from the MCMC chain output. Using the modal

assignments, the nodes are ordered by block label. This ordering applied to the log edge-weight matrix W and P in Figure 5a and Figure 5b respectively. The negative binomial model is more flexible and is thus able to more easily detect structure in the network compared to the Poisson model. This is exemplified in the ordered plot of the log edge weights in Figures 5a and 6a. Furthermore, the fit using the Poisson distribution for edge weights finds one large group (fourth from the left in Figure 5b) with a low incidence of sent emails. This group corresponds to parameter λ_4, which has a posterior mode of 0.19. Under the negative binomial distribution, the low-incidence group is much smaller, with modal parameters $r_9 = 0.004$ and $p_9 = 0.012$ giving an expected number of emails sent by a node in block nine as $r(1-p)/p \simeq 0.33$. The modal parameter values for each model are given in Table 6 together with the 5% and 95% quantiles.



(a) Edge weights $\log(1+W)$, permuted by the posterior modal block membership

(b) Joint posterior probability of belonging to same block P

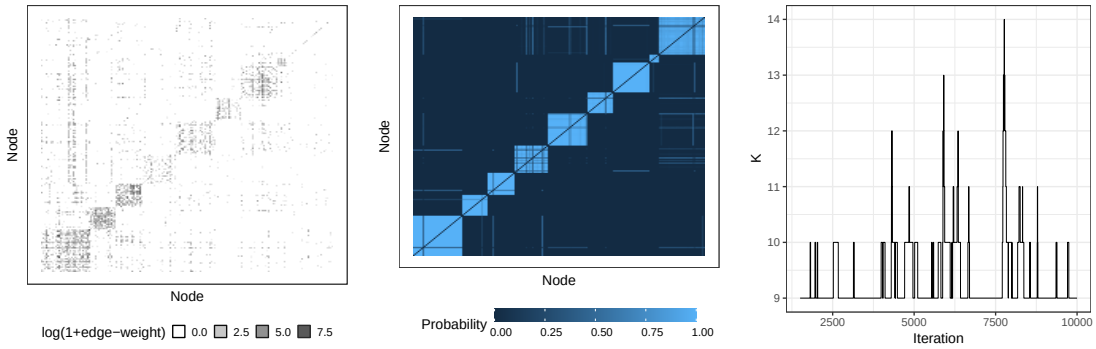
(c) Trace plot for K

Figure 5: Posterior summaries for block membership in Enron network with Poisson edge-weight model (after burn-in).

6. Concluding remarks

This paper considered a generalisation of the stochastic block model by allowing arbitrary edge-weight distributions and explicitly modelling the number of blocks. A Bayesian inference algorithm was proposed: a split-merge reversible jump Markov chain Monte Carlo sampler as described in Section 3. Unlike previous Bayesian treatments of the stochastic block model with an unknown number of blocks (Mørup and Schmidt, 2012, 2013; McDaid et al., 2013), the proposed algorithm handles edge-weight distributions without conjugate priors. This allows for more flexible modelling of network data, as demonstrated in Section 5.2 on the Enron email network. In this example, a negative binomial model (with both parameters unknown) was fit to the edge weights, allowing for a higher variance of edge weights within a block than under the Poisson model. In the Enron data set, the negative binomial explored the parameter space better than the Poisson model since it visited posterior states with more structure.

The algorithm presented here is general and can be applied to the generalised stochastic block model with any edge-weight distributions from which samples can be taken and densities evaluated. This can easily include co-variate information in either the edge-weight distribution, G, or the block membership distribution, F.



(a) Edge weights $\log(1+W)$, permuted by the posterior modal block membership

(b) Joint posterior probability of belonging to same block P

(c) Trace plot for K

Figure 6: Posterior summaries for block membership in Enron network with negative binomial edge-weight model (after burn-in).

For simplicity, the models presented in Section 2 assume all edges are present in the network and that each edges has a recorded edge weight. This assumption can be relaxed in (at least) two ways. Firstly, if some set of edges $\mathcal{A}$ is known to be absent from the network, then the set of edges is $\mathcal{E}_A = \mathcal{E}/\mathcal{A}$. For example, consider a network of electrical cables between substations. The substations are represented by nodes, the cables by edges and the voltage along a cable by an edge weight. In this case, Equation 2 remains unchanged except the last line runs over all $ij \in \mathcal{E}_A$ rather than $\mathcal{E}$. To adapt the split-merge sampler, the likelihood calculations involving node i iterate over all nodes $j \in \mathcal{E}_A/\{i\}$ instead of all $i \neq j$. In the second case, the edge exists in the model but the edge weight is not recorded in the data set; this is a missing data problem. Two approaches are possible: either the edge weight was not recorded, or the edge does not exist. In the first case, one could use a data augmentation scheme within the split-merge sampler to infer the state of missing edge weights. In the second case, a sparsity parameter as in Matias and Miele (2017) could be inferred within the GSBM framework. This treats edge weights as a mixture of the density G and a Dirac mass at zero representing the non-existence of an edge.

Acknowledgements

The author would like to thank the referees as well as Brendan Murphy, Simon Lunagomez and Peter Neal for helpful comments. Funding: This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) [EP/H023151/1 and EP/P033075/1].

Appendix A. Acceptance probability calculations

Since a merge move is the inverse of a split move, $A_{merge} = 1/A_{split}$, hence only A_{split} is derived. The acceptance probability can be split into the following parts: posterior density ratio, proposal density ratio, ratio of densities of auxiliary variables, and the Jacobian; as such A_{split} has the

Table 6: Parameter mode, 5% and 95% posterior quantiles for the Enron data with edge-weight model: (i) Possion(λ) and (ii) NegativeBinomial(r, p)

θ	Mode	5%	95%
r_0	0.012	0.011	0.012
r_1	0.133	0.122	0.147
r_2	0.323	0.282	0.374
r_3	0.169	0.149	0.194
r_4	0.086	0.069	0.106
r_5	0.082	0.070	0.100
r_6	0.114	0.092	0.139
r_7	0.120	0.104	0.137
r_8	0.460	0.259	0.706
r_9	0.004	0.002	0.022
p_0	0.013	0.012	0.015
p_1	0.003	0.002	0.003
p_2	0.007	0.006	0.009
p_3	0.002	0.002	0.003
p_4	0.020	0.014	0.029
p_5	0.007	0.005	0.010
p_6	0.008	0.005	0.011
p_7	0.006	0.005	0.008
p_8	0.039	0.019	0.064
p_9	0.012	0.001	0.041
λ_0	1.45	1.39	1.50
λ_1	43.67	41.49	45.29
λ_2	32.43	30.33	34.70
λ_3	52.62	51.69	57.98
λ_4	0.19	0.15	0.23
λ_5	30.28	27.15	31.14
λ_6	146.85	142.71	151.71
λ_7	498.32	492.65	505.24
λ_8	29.51	20.73	174.72
λ_9	161.93	23.59	343.92

general form:

$$\begin{aligned} A_{split} &= \frac{\pi(\kappa+1, \boldsymbol{z}', \boldsymbol{\theta}'|E)}{\pi(\kappa, \boldsymbol{z}, \boldsymbol{\theta}|E)} \frac{\mathrm{q}(\kappa, \boldsymbol{z}, \boldsymbol{\theta}|\kappa+1, \boldsymbol{z}', \boldsymbol{\theta}')}{\mathrm{q}(\kappa+1, \boldsymbol{z}', \boldsymbol{\theta}'|\kappa, \boldsymbol{z}, \boldsymbol{\theta})} \frac{\mathrm{q}(\lambda)}{\mathrm{q}(u', \lambda')} J_{split} \\ &= \frac{\pi(\kappa+1, \boldsymbol{z}', \boldsymbol{\theta}'|E)}{\pi(\kappa, \boldsymbol{z}, \boldsymbol{\theta}|E)} \frac{\mathrm{q}(merge|\kappa+1)}{\mathrm{q}(split|\kappa)} \frac{\mathrm{q}(k', l')}{\mathrm{q}(k)} \frac{\mathrm{q}(\lambda)}{\mathrm{q}(\lambda', u')} \frac{1}{\mathrm{q}(\boldsymbol{z}'|\boldsymbol{\theta}')} J_{split} \end{aligned} \tag{A.1}$$

where $\mathrm{q}(split|\kappa)$ and $\mathrm{q}(merge|\kappa)$ are the probabilities of proposing a split or merge move given that the current state of the sampler contains κ blocks. These are chosen as $1/2$ where possible. That is $\mathrm{q}(split|\kappa = 1) = 1$ and $\mathrm{q}(merge|\kappa = 1) = 0$ since merging is impossible when there is only one block. Note that in the examples: $\lambda, \lambda' \overset{\text{iid}}{\sim} \mathrm{Unif}(0, 1)$, $u' \sim \mathrm{Normal}(0, 1)$, k' and k, l are sampled at

random amongst the set of available blocks.

Finally, J_{split} is the Jacobian of the split proposal given in Equation (A.2) and p is the dimensionality of each $\boldsymbol{\theta}_k$.

$$J_{split} = \begin{vmatrix} \frac{\partial \boldsymbol{\theta}'_{k'}}{\partial \boldsymbol{\theta}_k} & \frac{\partial \boldsymbol{\theta}'_{l'}}{\partial \boldsymbol{\theta}_k} \\ \frac{\partial \boldsymbol{\theta}'_{k'}}{\partial u'} & \frac{\partial \boldsymbol{\theta}'_{l'}}{\partial u'} \end{vmatrix} = \left| \frac{\nabla \mathrm{m}(\boldsymbol{\theta}'_{k'}) \nabla \mathrm{m}(\boldsymbol{\theta}'_{l'})}{\nabla \mathrm{m}(\boldsymbol{\theta}_k) (2\lambda(1-\lambda))^p} \right| \tag{A.2}$$

Therefore, in the examples, where specific choices for u', λ', λ and $\mathrm{q}(merge), \mathrm{q}(split)$ have been made, the acceptance probabilities reduce to:

$$\begin{aligned} A_{split} =& \frac{\pi(\kappa+1, \boldsymbol{z}', \boldsymbol{\theta}'|E)}{\pi(\kappa, \boldsymbol{z}, \boldsymbol{\theta}|E)} \frac{1}{1+\mathbb{I}[\kappa=1]} \frac{2}{\kappa+1} \\ & \times \frac{1}{\phi(u'|0,\sigma^2)} \frac{1}{\mathrm{q}(\boldsymbol{z}'|\boldsymbol{\theta}')} \left| \frac{\nabla \mathrm{m}(\boldsymbol{\theta}'_{k'}) \nabla \mathrm{m}(\boldsymbol{\theta}'_{l'})}{\nabla \mathrm{m}(\boldsymbol{\theta}_k) (2\lambda(1-\lambda))^p} \right| \\ A_{merge} =& \frac{\pi(\kappa-1, \boldsymbol{z}', \boldsymbol{\theta}'|E)}{\pi(\kappa, \boldsymbol{z}, \boldsymbol{\theta}|E)} \left(1+\mathbb{I}[\kappa=2]\right) \frac{\kappa}{2} \\ & \times \phi\big(u|0,\sigma^2\big) \mathrm{q}(\boldsymbol{z}|\boldsymbol{\theta}) \left| \frac{\nabla \mathrm{m}(\boldsymbol{\theta}'_{k'}) (2\lambda(1-\lambda))^p}{\nabla \mathrm{m}(\boldsymbol{\theta}_k) \nabla \mathrm{m}(\boldsymbol{\theta}_l)} \right| \end{aligned}$$

Appendix B. Post-hoc matching

The GSBM is invariant to relabelling of the nodes – Equation 4 gives the same posterior value if the node labels are permuted. This causes a problems when comparing the output of the MCMC against some known parameter values in Section 4, since the estimated block labels need to match the truth for a reasonable comparison.

Let Z^{true} be a set of true block labels. We match the MCMC output labels to the true labels by matching the modal assignment vector Z^{mode} to Z^{true}, where

$$Z_i^{\text{mode}} = \arg\max_k \sum_{\mathcal{S}} \mathbb{I}[Z_{is} = k],$$

gives the most-often used block label for node i during the MCMC iterations in $\mathcal{S}$.

Given Z^{true} and Z^{mode}, a contingency table n is formed via:

$$n_{ck} = \sum_i \mathbb{I}\big[(Z_i^{\text{mode}} = c) \& (Z_i^{\text{true}} = k)\big].$$

Thus entry c, k in the table is the number of nodes assigned to block c under the mode and block k under the truth.

Let π be a permutation with $\pi_c = \arg\max_k n_{ck}$. We relabel the MCMC output for each $i = 1, \ldots, N$ and $s \in \mathcal{S}$ via $Z_{is} = c \mapsto Z_{is} = \pi_c$ and $\theta_c \mapsto \theta_{\pi_c}$. Under this relabelling the modal and true labels match so comparisons between parameters can be made.

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., Xing, E. P., 2008. Mixed membership stochastic blockmodels. Journal of Machine Learning Research 9 (Sep), 1981–2014.

Ambroise, C., Matias, C., 2012. New consistent and asymptotically normal parameter estimates for random-graph mixture models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 74 (1), 3–35.

Chen, K., Lei, J., 2016. Network cross-validation for determining the number of communities in network data. Journal of the American Statistical Association, 1–11.
URL https://doi.org/10.1080/01621459.2016.1246365

Copic, J., Jackson, M. O., Kirman, A., 2009. Identifying community structures from network data via maximum likelihood methods. The BE Journal of Theoretical Economics 9 (1).

Daudin, J.-J., Picard, F., Robin, S., 2008. A mixture model for random graphs. Statistics and Computing 18 (2), 173–183.
URL https://doi.org/10.1007/s11222-007-9046-7

Fienberg, S. E., Meyer, M. M., Wasserman, S. S., 1985. Statistical analysis of multiple sociometric relations. Journal of the american Statistical association 80 (389), 51–67.

Frank, O., Harary, F., 1982. Cluster inference by using transitivity indices in empirical graphs. Journal of the American Statistical Association 77 (380), 835–840.

Gelman, A., Rubin, D. B., 1992. Inference from iterative simulation using multiple sequences. Statist. Sci. 7 (4), 457–472.
URL https://doi.org/10.1214/ss/1177011136

Geng, J., Bhattacharya, A., Pati, D., 2019. Probabilistic community detection with unknown number of communities. Journal of the American Statistical Association 114 (526), 893–905.

Gershman, S. J., Blei, D. M., 2012. A tutorial on Bayesian nonparametric models. Journal of Mathematical Psychology 56 (1), 1 – 12.
URL https://doi.org/10.1016/j.jmp.2011.08.004

Green, P. J., 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 82 (4), 711–732.

Green, P. J., Richardson, S., 2001. Modelling heterogeneity with and without the Dirichlet process. Scandinavian Journal of Statistics 28 (2), 355–375.
URL http://dx.doi.org/10.1111/1467-9469.00242

Hoff, P. D., Raftery, A. E., Handcock, M. S., 2002. Latent space approaches to social network analysis. Journal of the American Statistical Association 97 (460), 1090–1098.
URL https://doi.org/10.1198/016214502388618906

Holland, P. W., Laskey, K. B., Leinhardt, S., 1983. Stochastic blockmodels: First steps. Social networks 5 (2), 109–137.

Jiang, Q., Zhang, Y., Sun, M., 2009. Community detection on weighted networks: A variational Bayesian method. In: Asian Conference on Machine Learning. Springer, pp. 176–190.

Karrer, B., Newman, M. E., 2011. Stochastic blockmodels and community structure in networks. Physical Review E 83 (1), 016107.

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., Ueda, N., 2006. Learning systems of concepts with an infinite relational model. In: AAAI. Vol. 3. p. 5.

Klimt, B., Yang, Y., 2004. Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, Ch. The Enron Corpus: A New Dataset for Email Classification Research, pp. 217–226.

Latouche, P., Birmele, E., Ambroise, C., 2012. Variational Bayesian inference and complexity control for stochastic block models. Statistical Modelling 12 (1), 93–115.

Lei, J., 2016. A goodness-of-fit test for stochastic block models. The Annals of Statistics 44 (1), 401–424.
URL https://doi.org/10.1214/15-aos1370

Ludkin, M., 2020. SBMSplitMerge: Inference for a Generalised SBM with a Split Merge Sampler. R package version 1.1.1.
URL https://cran.r-project.org/package=SBMSplitMerge

Ludkin, M., Eckley, I., Neal, P., 2018. Dynamic stochastic block models: parameter estimation and detection of changes in community structure. Statistics and Computing.
URL https://doi.org/10.1007/s11222-017-9788-9

Mariadassou, M., Robin, S., Vacher, C., 2010. Uncovering latent structure in valued graphs: a variational approach. The Annals of Applied Statistics, 715–742.

Matias, C., Miele, V., 2017. Statistical clustering of temporal networks through a dynamic stochastic block model. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79 (4), 1119–1141.

Matias, C., Robin, S., 2014. Modeling heterogeneity in random graphs through latent space models: a selective review. ESAIM: Proc. 47, 55–74.
URL https://doi.org/10.1051/proc/201447004

McDaid, A. F., Murphy, T. B., Friel, N., Hurley, N. J., 2013. Improved Bayesian inference for the stochastic block model with application to large networks. Computational Statistics & Data Analysis 60, 12–31.
URL http://dx.doi.org/10.1016/j.csda.2012.10.021

Miller, J. W., Harrison, M. T., 2018. Mixture models with a prior on the number of components. Journal of the American Statistical Association 113 (521), 340–356.

Mørup, M., Schmidt, M. N., 2012. Bayesian community detection. Neural computation 24 (9), 2434–2456.

Mørup, M., Schmidt, M. N., 2013. Nonparametric Bayesian modeling of complex networks: an introduction. IEEE Signal Processing Magazine 30 (3), 110–128.

Mørup, M., Schmidt, M. N., Hansen, L. K., 2011. Infinite multiple membership relational modeling for complex networks. In: Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on. IEEE, pp. 1–6.

Négyessy, L., Nepusz, T., Kocsis, L., Bazsó, F., 2006. Prediction of the main cortical areas and connections involved in the tactile function of the visual cortex by network analysis. European Journal of Neuroscience 23 (7), 1919–1930.

Nobile, A., Fearnside, A. T., 2007. Bayesian finite mixtures with an unknown number of components: The allocation sampler. Statistics and Computing 17 (2), 147–162.
URL https://doi.org/10.1007/s11222-006-9014-7

Nowicki, K., Snijders, T. A. B., 2001. Estimation and prediction for stochastic blockstructures. Journal of the American Statistical Association 96 (455), 1077–1087.
URL https://doi.org/10.1198/016214501753208735

Peixoto, T. P., 2013. Parsimonious module inference in large networks. Physical Review Letters 110 (14).
URL https://doi.org/10.1103/physrevlett.110.148701

Saldaña, D. F., Yu, Y., Feng, Y., 2017. How many communities are there? Journal of Computational and Graphical Statistics 26 (1), 171–181.
URL https://doi.org/10.1080/10618600.2015.1096790

Snijders, T. A., Nowicki, K., 1997. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. Journal of classification 14 (1), 75–100.

Snijders, T. A. B., Pattison, P. E., Robins, G. L., Handcock, M. S., 2006. New specifications for exponential random graph models. Sociological Methodology 36 (1), 99–153.

Wang, Y. R., Bickel, P. J., et al., 2017. Likelihood-based model selection for stochastic block models. The Annals of Statistics 45 (2), 500–528.

Wasserman, S., Anderson, C., 1987. Stochastic a posteriori blockmodels: Construction and assessment. Social Networks 9 (1), 1–36.
URL https://doi.org/10.1016/0378-8733(87)90015-3

Xin, L., Zhu, M., Chipman, H., 2017. A continuous-time stochastic block model for basketball networks. The Annals of Applied Statistics 11 (2), 553–597.