Electronic Journal of Statistics
Vol. 15 (2021) 1814–1868
ISSN: 1935-7524
https://doi.org/10.1214/21-EJS1819

Sparse networks with core-periphery structure*

Cian Naik

Department of Statistics, University of Oxford
e-mail: cian.naik@stats.ox.ac.uk

François Caron

Department of Statistics, University of Oxford
e-mail: caron@stats.ox.ac.uk

Judith Rousseau

Department of Statistics, University of Oxford
e-mail: judith.rousseau@stats.ox.ac.uk

Abstract: We propose a statistical model for graphs with a core-periphery structure. We give a precise notion of what it means for a graph to have this structure, based on the sparsity properties of the subgraphs of core and periphery nodes. We present a class of sparse graphs with such properties, and provide methods to simulate from this class, and to perform posterior inference. We demonstrate that our model can detect core-periphery structure in simulated and real-world networks.

MSC2020 subject classifications: Primary 62F15, 05C80; secondary 60G55.
Keywords and phrases: Bayesian nonparametrics, completely random measures, Poisson random measures, networks, random graphs, sparsity, point processes.

Received October 2019.

1. Introduction

Network data arises in a number of areas, including social, biological and transport networks. Modelling of this type of data has become an increasingly important field in recent times, partly due to the greater availability of the data, and also due to the fact that we increasingly want to model complex systems with many interacting components. A central topic within this field is the design of random graph models. Various models have been proposed, building on the early work of Erdős [13]. A key goal with these models is to capture important properties of real-world graphs. In particular, a lot of effort has gone into the

*Cian Naik has been supported by the Engineering and Physical Sciences Research Council and Medical Research Council [award reference 1930478]. François Caron has been supported by the Engineering and Physical Sciences Research Council under grant EP/P026753/1. Judith Rousseau has received funding for the project from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 834175).

investigation of meso-scale structures [39]. These are intermediate scale structures such as communities of nodes within the overall network. There has been a large amount of work devoted to models that can capture community structure, such as the popular stochastic block-model [20, 40, 35].

Here, we focus on a different type of meso-scale structure, known as a core-periphery structure, which we will define formally in Section 2.1. The intuitive idea is of a network that consists of two classes of nodes — *core* nodes that are densely connected to each other, and *periphery* nodes that are more loosely connected to core nodes, and sparsely connected between themselves.

The network of flight connections between airports is a typical example of a core-periphery network. The core nodes correspond to central hubs that flights are routed through, according to the so-called Spoke-hub distribution [21], while other airports are more sparsely connected between each other. Other examples include the World Wide Web [9] and social [42], biological [29], transport [21], citation [4], trade [32] or financial [15] networks. The importance of being able to identify core-periphery structure can be seen by considering the properties of these networks. In cases such as transport networks, this identification allows for the detection of hubs which may be the most important locations for additional development. In protein-protein interaction networks, identifying a core helps to determine which proteins are the most important for the development of the organism. In internet networks, core nodes could be the most important places to defend from cyber-attacks. In this work, we are particularly interested in models for sparse graphs[1]. The advantage of our core-periphery model is that the statistical network modelling framework that we work in allows us to capture the sparsity that many of these networks have [2, 36].

A number of algorithmic approaches have been proposed for the detection of core-periphery structure [12, 10, 11], see [39] for a review. [43] recently developed a class of sparse graphs with locally dense subgraphs. The construction is based on Poisson random measures as in our case, but the objective and properties of both models are rather different. The graphs of [43] have a growing number of dense subgraphs, where each subgraph has a bounded number of nodes, and no subgraph is identified as a periphery. In contrast, our model has a single dense core, whose size is unbounded, and a sparse periphery.

An alternative, popular, model-based approach is to consider a two-group stochastic blockmodel [44] where

$$\Pr(z_{ij} = 1 \mid \Pi, c_i, c_j) = \Pi_{c_i,c_j}$$

where $c_i = 1$ if node i is in the core and $c_i = 2$ otherwise,

$$\Pi = \begin{pmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{12} & \Pi_{22} \end{pmatrix}$$

with Π_{11}, Π_{12} and Π_{22} respectively the core-core, core-periphery and periphery-periphery probabilities of connection, where typically Π_{12}, Π_{22} are far smaller

[1] A graph is said to be sparse if the number of edges scales subquadratically with the number of nodes. Otherwise, it is said to be dense.

than Π_{11}. However, such models are known to generate dense graphs. To obtain sparse graph sequences in the stochastic blockmodel setting, one can have the matrix Π to depend on n, e.g. $\Pi_n = \frac{\Pi_1}{n}$ for some initial matrix Π_1 [44]. However in this case, the graph family is not projective any more for different network sizes. The approach considered in this paper allows to have both a sparse and projective graph family.

The contributions of our work are as follows. Firstly, we provide a precise definition of what it means for a graph to have a core-periphery structure, based on the sparsity properties of the subgraphs of core and periphery nodes. Secondly, building on earlier work from [41], we present a class of sparse network models with such properties, and provide methods to simulate from this class, and to perform posterior inference. Finally, we demonstrate that our approach can detect meaningful core-periphery structure in two real-world airport and trade networks, while providing a good fit to global structural properties of the networks.

The statistical network modelling framework our work is set in, introduced by [6], is based on representing the network as an exchangeable random measure. Sparse graph models with (overlapping) block structure have already been proposed within this framework [19, 41]. However, these models cannot be applied directly to model networks with a core-periphery structure, as they make the assumption that a single parameter tunes the overall sparsity properties of the graph, with the same structural sparsity properties across different blocks. This is an undesirable property for core-periphery networks, where the subgraph of core nodes is expected to have different, denser structural properties than the rest of the network. Our work builds on the framework of (multivariate) completely random measures (CRMs) [23, 24] that have been widely used in the Bayesian nonparametric literature [38, 27, 17] to construct graphs with heterogeneous sparsity properties.

The rest of the paper is organised as follows. In Section 2 we define what a core-periphery structure means in our framework, and give the general construction of this type of network as well as the particular model that we employ. We show how our framework can accommodate both core-periphery and community structure. In Section 3 we present some important theoretical results about the new model, such as the sparsity properties that define the core-periphery structure. In Section 4 we give the details of the Markov Chain Monte Carlo (MCMC) sampler used to perform posterior inference, and test our model on a variety of simulated and real data sets. We also compare it against a range of contemporary alternatives, and show that it provides an improvement in certain settings.

Notations. We follow the asymptotic notations of [22]. Let $(X_\alpha)_{\alpha\geq 0}$ and $(Y_\alpha)_{\alpha\geq 0}$ be two stochastic processes defined on the same probability space with $X_\alpha, Y_\alpha \to \infty$ almost surely (a.s.) as $\alpha \to \infty$. We have $X_\alpha = O(Y_\alpha)$ a.s. $\iff$ $\limsup_{\alpha\to\infty} \frac{X_\alpha}{Y_\alpha} < \infty$ a.s.; $X_\alpha = o(Y_\alpha)$ a.s. $\iff$ $\limsup_{\alpha\to\infty} \frac{X_\alpha}{Y_\alpha} = 0$ a.s.; $X_\alpha = \Theta(Y_\alpha)$ a.s. $\iff$ $X_\alpha = O(Y_\alpha)$ and $Y_\alpha = O(X_\alpha)$ a.s.

2. Statistical network models with core-periphery structure

2.1. Definitions

In this section, we formally define what it means for graphs to be sparse and to have a core-periphery structure.

Let $G = (G_\alpha)_{\alpha \geq 0}$ be a family of growing undirected random graphs with no isolated vertices, where $\alpha \geq 0$ is interpreted as a size parameter. $G_\alpha = (V_\alpha, E_\alpha)$ where V_α and E_α are the set of vertices and edges respectively. Denote respectively $N_\alpha = |V_\alpha|$ and $N_\alpha^{(e)} = |E_\alpha|$ for the number of nodes and edges in G_α, and assume $N_\alpha, N_\alpha^{(e)} \to \infty$ almost surely as $\alpha \to \infty$.

We first give the definition of sparsity for the family $G = (G_\alpha)_{\alpha \geq 0}$.

Definition 2.1 (Dense and Sparse graphs)**.** We say that a graph family G is *dense* if the number of edges scales quadratically with the number of nodes

$$N_\alpha^{(e)} = \Theta(N_\alpha^2) \tag{2.1}$$

almost surely as $\alpha \to \infty$. Conversely, it is *sparse* if the number of edges scales subquadratically with the number of nodes

$$N_\alpha^{(e)} = o(N_\alpha^2) \tag{2.2}$$

almost surely as $\alpha \to \infty$.

Let $(V_{\alpha,c})_{\alpha \geq 0}$ be a growing family of core nodes, where $V_{\alpha,c} \subseteq V_\alpha$ for all α. Let $N_{\alpha,c} = |V_{\alpha,c}|$ be the number of core nodes, and $N_{\alpha,c-c}^{(e)}$ the number of edges between nodes in the core. Assume both $N_{\alpha,c}, N_{\alpha,c-c}^{(e)} \to \infty$ almost surely as $\alpha \to \infty$.

Definition 2.2 (Core-periphery structure)**.** We say that a graph family $G = (G_\alpha)_{\alpha \geq 0}$ is sparse with core-periphery structure if the graph is sparse with a dense core subgraph, that is

$$N_{\alpha,c-c}^{(e)} = \Theta(N_{\alpha,c}^2) \quad N_\alpha^{(e)} = o(N_\alpha^2). \tag{2.3}$$

A consequence of (2.3) is that $N_{\alpha,c} = o(N_\alpha)$, since $N_{\alpha,c-c}^{(e)} \leq N_\alpha^{(e)}$. In other words, the core corresponds to a small dense subgraph of a sparse graph, with sparse connections to the other part of the graph, which is called the periphery.

2.2. A model for networks with core-periphery structure

Having defined what we mean by core-periphery structure, we now give a construction of a core-periphery network. Building on the model of [41], we also allow the network to exhibit community structure. Following [6], we represent a graph by the point process on the plane

$$Z = \sum_{i,j} z_{ij} \delta_{(\theta_i, \theta_j)} \tag{2.4}$$

where $z_{ij} = z_{ji} = 1$ if i and j are connected, and 0 otherwise. Here, each node i is located at some point $\theta_i \in \mathbb{R}_+ = [0, \infty)$. A finite graph G_α of size $\alpha > 0$ is obtained by considering the restriction of Z to $[0, \alpha]^2$, see [6]. As in [41], we consider that the probability of a connection between nodes i and j is given by the link function

$$\Pr(z_{ij} = 1 | (w_{l1}, w_{l2}, \dots, w_{lK})_{l=1,2,\dots}) = \begin{cases} 1 - e^{-2\sum_{k=1}^K w_{ik} w_{jk}} & i \neq j \\ 1 - e^{-\sum_{k=1}^K w_{ik}^2} & i = j \end{cases} \tag{2.5}$$

Here, $w_{i1} \geq 0$ is the core parameter and the parameters $w_{ik} > 0$, for $k = 2, \dots, K$, are interpreted as the degree of affiliation of node i to community k. In the particular case $K = 2$, we have core-periphery but no community structure.

The model parameters $(w_{i1}, w_{i2}, , \dots, w_{iK}, \theta_i)_{i=1,2,\dots}$ are the points of a Poisson point process on $\mathbb{R}_+^{K+1}$ with mean measure $\rho(dw_1, dw_2, \dots, dw_K)d\theta$ where ρ is a σ-finite measure on $\mathbb{R}_+^K$, concentrated on $\mathbb{R}_+^K \backslash \{(0, 0, \dots, 0)\}$, which satisfies $\int_{\mathbb{R}_+^K} \min(1, \sum_{k=1}^K w_k) \rho(dw_1, dw_2, \dots, dw_K) < \infty$. As shown in [41], the resulting graph is sparse if

$$\int_{[0,\infty)^K} \rho(dw_1, dw_2, \dots, dw_K) = \infty, \tag{2.6}$$

and dense otherwise. [41] considered a specific class of models for ρ, based on compound completely random measures (CRMs) [17], where ρ is concentrated on $(0, \infty)^K$; that is, for all nodes i and communities k we have $w_{ik} > 0$. As a consequence, the sparsity pattern is homogeneous across the graph. We consider a different and more flexible construction here where $w_{i1} \geq 0$ and $w_{ik} > 0,\ k = 2, \dots, K$. As will be shown in Theorem 3.1, the core-periphery property can be enforced by making the following assumptions on the mean measure ρ:

$$\int_{(0,\infty)\times\{0\}^{K-1}} \rho(dw_1, dw_2, \dots, dw_K) = 0, \tag{2.7}$$

$$\int_{\{0\}\times(0,\infty)^{K-1}} \rho(dw_1, dw_2, \dots, dw_K) > 0, \tag{2.8}$$

$$0 < \int_{(0,\infty)^K} \rho(dw_1, dw_2, \dots, dw_K) < \infty. \tag{2.9}$$

We identify the set $\mathcal{C} = \{i \mid w_{i1} > 0\}$ as the core nodes, and the remaining ones $\mathcal{P} = \{i \mid w_{i1} = 0\}$ as the periphery nodes. Assumption (2.7) ensures that we do not have a subset of core nodes disconnected from the rest of the graph. Along with Equation (2.8), it also ensures that each node has at least one strictly positive community affiliation parameter. In practice, we use a construction where all of the community affiliation parameters are strictly positive.

The strict positivity assumptions in Equations (2.8) and (2.9) ensure that the size of the core or periphery is not empty with probability 1. The boundedness assumption in Equation (2.9) ensures that the subgraph of core nodes is dense, as will be shown in Theorem 3.1. Note that the overall graph may be sparse or dense, depending whether the integral in Equation (2.8) is finite or not.

We now propose a way to construct such a function ρ. We start with a *base* Lévy measure ρ_0 on $(0,\infty)$. Let $(w_{i0},\theta_i)_{i\geq 1}$ be the points of a Poisson point process with mean measure $\rho_0(dw_0)d\theta$ and set

$$\begin{aligned} w_{ik} &= \beta_{ik}w_{i0}, \quad k=1,2,\dots,K \\ \beta_{i1}\mid w_{i0} &\sim \text{Ber}(1-e^{-w_{i0}}), \quad (\beta_{i2},\dots,\beta_{iK}) \overset{iid}{\sim} F \end{aligned} \tag{2.10}$$

where F is a probability distribution on $\mathbb{R}_+^{K-1}$ and the scores $\beta_{i1}\in\{0,1\}$ and $(\beta_{i2},\dots,\beta_{iK})\in\mathbb{R}_+^{K-1}>0,\ i\geq 1$ are mutually independent. In other words, with probability $1-e^{-w_{i0}}$, $w_{i1}=w_{i0}>0$ and i is a core node, while with probability $e^{-w_{i0}}$, $w_{i1}=0$ and i belongs to the periphery. ρ is then given by:

$$\rho(dw_1,dw_2,\dots,dw_K) = \\ \int_0^\infty \left[\left((1-e^{-w_0})\delta_{w_0}(dw_1)+e^{-w_0}\delta_0(dw_1)\right)\frac{F\left(\frac{dw_2}{w_0},\dots,\frac{dw_K}{w_0}\right)}{w_0^{K-1}}\right]\rho_0(dw_0) \tag{2.11}$$

In this paper, we set ρ_0 to be the mean measure of the jump part of a generalized gamma process (GGP)

$$\rho_0(dw_0)=\frac{1}{\Gamma(1-\sigma)}w_0^{-1-\sigma}\exp(-w_0\tau)dw_0 \tag{2.12}$$

where (σ,τ) satisfy $\sigma\in(0,1),\tau\geq 0$ or $\sigma\in(-\infty,0],\tau>0$. The GGP has been used extensively due to its flexibility, the interpretability of its parameters and its conjugacy properties [26, 7].

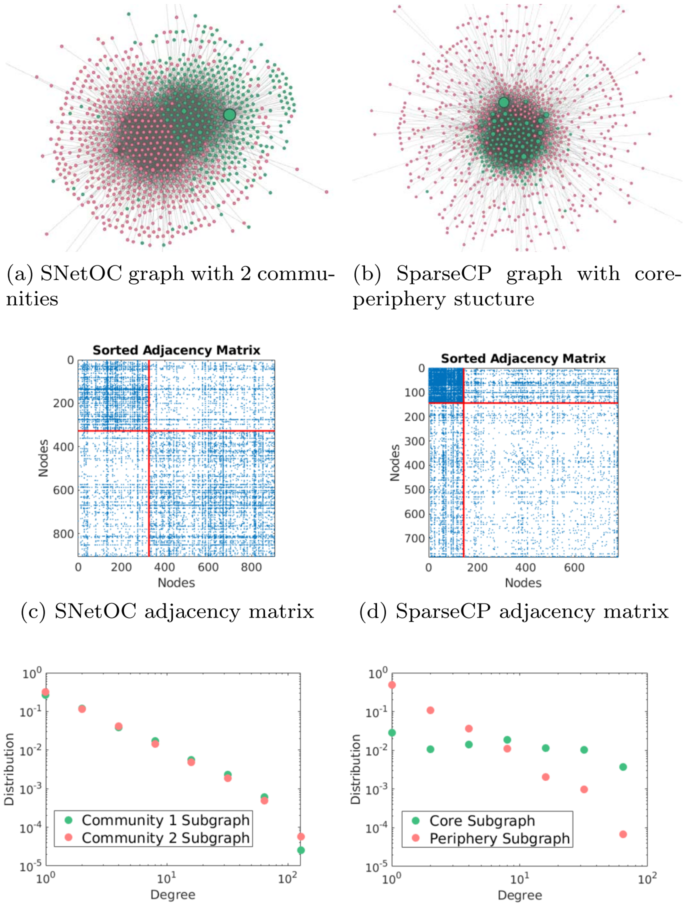
Note crucially that, although the formulation (2.10) resembles the formulation of the class of compound CRMs introduced by [17], in our construction, the scores β_{i1} are not identically distributed, and depend on the base parameter w_{i0}. This key difference enables the model to have different sparsity properties, as shown in Section 3.

Of particular interest, for computational reasons, is the case where F is set to be the product of independent gamma distributions

$$F(d\beta_2,\dots,d\beta_K)=\prod_{k=2}^K \beta_k^{a_k-1}e^{-b_k\beta_k}\frac{b_k^{a_k}}{\Gamma(a_k)}d\beta_k, \tag{2.13}$$

where $a_k>0$ and $b_k>0$ are hyperparameters for $k=1,\dots,K$.

We simulate a network from the above model with parameters $K=2,\alpha=200,\sigma=0.2,\tau=1,b=0.5,a=0.2$. This corresponds to the case where we have core-periphery but not community structure. In either case, we refer to our model as the Sparse Core Periphery (SparseCP) model. For comparison, we also simulate a network from the Sparse Network with Overlapping Communities (SNetOC) model of [41] with $K=2$ communities and parameters $\alpha=100,\sigma=$



(a) SNetOC graph with 2 communities

(b) SparseCP graph with core-periphery stucture

(c) SNetOC adjacency matrix

(d) SparseCP adjacency matrix

(e) SNetOC subgraph degree distributions

(f) SparseCP subgraph degree distributions

Fig 1. *Comparing a core-periphery graph with no community structure simulated from the SparseCP model to a graph with two communities from the SNetOC model of [41]. Plots of (a) the graph with two communities, with nodes from each community in green and red respectively, and (b) the core-periphery graph with core nodes in green and periphery nodes in red. In each case the size of the nodes is proportional to their mean sociability. Adjacency matrices for (c) the graph with two communities and (d) the core-periphery graph. In each case the nodes are ordered by community or core membership, with the red lines partitioning the two groups. Degree distributions of (e) the subgraphs of nodes within communities 1 and 2 for the two community graph, and (f) the subgraphs of core and periphery nodes for the core-periphery graph.*

$0.2, \tau = 2.5, b = (0.2, 0.05), a = (0.4, 0.1)$. The parameters are chosen so that the graphs have similar sizes and overall degree distributions.

Figures 1(a) and 1(b) show the networks sampled from both models. Figures 1(c) and 1(d) show the associated adjacency matrices, where nodes are

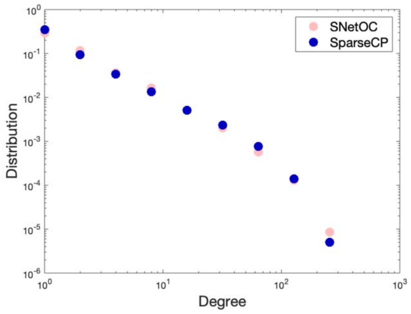


FIG 2. *Overall degree distribution of the SNetOC and SparseCP graphs.*

ordered according to the community where they have the largest affiliation for the SNetOC model, and by core ($w_{i1} > 0$) or periphery ($w_{i1} = 0$) affiliation for the SparseCP model. In Figures 1(e) and 1(f) we represent the empirical degree distributions of nodes within each community for SNetOC, and for the core and periphery subgraphs for the SparseCP model. In these plots, and all subsequent degree distribution plots, the degrees have been binned together in evenly spaced logarithmic bins, see e.g. [34]. Although the overall degree distributions for both models are very close (see Figure 2), the degree distributions of the subgraphs are very different: the degree distributions within each subgraph have a similar power-law behaviour for SNetOC, whereas for the SparseCP model, it exhibits a Poisson-like behaviour for the core, and a power-law behaviour for the periphery.

2.2.1. Parameter interpretability

In summary, the different parameters in the model control the structure of the network as follows:

- $\mathbf{w_{i1}}$ — We interpret w_{i1} as a *coreness* parameter, with $w_{i1} > 0$ indicating that node i is in the core.
- $\mathbf{w_{i2}}, \dots, \mathbf{w_{iK}}$ — When $K = 2$, we can interpret w_{i2} as an overall sociability parameter. Otherwise $w_{i2}, \dots, w_{iK}$ indicate the affiliation to each of the respective communities, and we can interpret them as sociabilities within each of these communities.
- $\boldsymbol{\sigma}$ — As we will see from Theorem 3.2, σ controls the overall sparsity of the network, with a higher value leading to sparser networks. It also controls the size of the core, with a larger value of σ leading to a smaller relative core size.
- $\boldsymbol{\tau}$ — As we can see from the theoretical results, τ does not affect the asymptotic rates for the densities of the different regions. Its main effect is to induce an exponential tilting of large degrees in the degree distribution, as we see in Figure 18 in Appendix B.

- $\mathbf{a_2}, \dots, \mathbf{a_K}, \mathbf{b_2}, \dots, \mathbf{b_K}$ — The hyperparameters of the community sociability parameters control the distribution F defined in Equation 2.13, which is the product of $\Gamma(a_k, b_k)$ distributions. Furthermore increasing the a_k decreases the relative size of the core by changing the sizes of the w_{ik} relative to w_{i1}, while increasing the b_k increases the relative size.
- $\boldsymbol{\alpha}$ — This is a size-parameter, tuning the number of nodes and edges in the network.

In the following section we show theoretically and through simulations that these models recover the core-periphery structure defined in Definition 2.2.

3. Core-periphery and sparsity properties

In this section we study the asymptotic behaviour of the number of nodes N_α, the number of nodes in the core $N_{\alpha,c}$ and in the periphery $N_{\alpha,p}$, together with the number of edges $N_\alpha^{(e)}$, the number of edges between core nodes $N_{\alpha,c-c}^{(e)}$, between periphery nodes $N_{\alpha,p-p}^{(e)}$ and between core and periphery nodes $N_{\alpha,c-p}^{(e)}$, which are the key quantities to understand the core periphery structure. They are defined as

$$\begin{aligned}
N_\alpha &= \sum_i \mathbb{1}_{\theta_i \le \alpha} \mathbb{1}_{(\sum_j z_{ij} \mathbb{1}_{\theta_j \le \alpha}) \ge 1}, \\
N_\alpha^{(e)} &= \sum_{i \le j} z_{ij} \mathbb{1}_{\theta_i \le \alpha} \mathbb{1}_{\theta_j \le \alpha} \\
N_{\alpha,c} &= \sum_i \mathbb{1}_{\theta_i \le \alpha} \mathbb{1}_{w_{i1} > 0} \mathbb{1}_{(\sum_j z_{ij} \mathbb{1}_{\theta_j \le \alpha}) \ge 1}, \\
N_{\alpha,c-c}^{(e)} &= \sum_{i \le j} z_{ij} \mathbb{1}_{w_{i1} > 0} \mathbb{1}_{w_{j1} > 0} \mathbb{1}_{\theta_i \le \alpha} \mathbb{1}_{\theta_j \le \alpha}
\end{aligned}$$

and similarly for the other quantities. In Theorem 3.1 we study the SparseCP model as defined by Equations (2.4) and (2.5), where the Lévy measure ρ satisfies Assumptions (2.7), (2.8) and (2.9).

Theorem 3.1. *Consider the graph family defined by Equations* (2.4) *and* (2.5), *where the Lévy measure ρ satisfies Assumptions* (2.7), (2.8) *and* (2.9). *Then the graph has a dense core subgraph*

$$N_{\alpha,c-c}^{(e)} = \Theta(N_{\alpha,c}^2) \textit{ almost surely as } \alpha \to \infty. \tag{3.1}$$

If

$$\int_{\{0\} \times (0,\infty)^{K-1}} \rho(dw_1, dw_2, \dots, dw_K) = \infty$$

then the graph is sparse overall and (2.2) *holds. Otherwise, it is dense, and* (2.1) *holds.*

We now characterize more precisely the sparsity properties for the particular model described by Equations (2.11), (2.12) together with (2.13), in Section 2.2.

Theorem 3.2. *Consider the graph family defined by Equations (2.4) and (2.5), where the Lévy measure ρ takes the form of Equation (2.11), with a generalized gamma process base measure ρ_0, and F a product of independent gamma distributions. Assume further that $\sigma \in (0,1)$ and $\tau > 0$, so that*

$$\int_0^\infty \rho_0(dw_0) = \infty, \qquad \int_0^\infty w_0 \rho_0(dw_0) < \infty. \tag{3.2}$$

Then, almost surely as α tends to infinity,

$$\begin{aligned} N_\alpha^{(e)} &= O\left(N_\alpha^{\frac{2}{1+\sigma}}\right), \quad & N_{\alpha,c-c}^{(e)} &= \Theta\left(N_{\alpha,c}^2\right) \\ N_{\alpha,p-p}^{(e)} &= O\left(N_{\alpha,p}^{\frac{2}{1+\sigma}}\right), \quad & N_{\alpha,c-p}^{(e)} &= O\left(\left(\sqrt{N_{\alpha,c}N_{\alpha,p}}\right)^{\frac{2}{1+\sigma/2}}\right). \end{aligned} \tag{3.3}$$

A natural consequence of the definition of core-periphery structure that we use is that the relative size of the core tends to zero, since the overall graph must be sparse. In Corollary 3.2.1 we confirm this, noting that $\sigma \in (0,1)$.

Corollary 3.2.1. *In the same setting as Theorem 3.2, we have that*

$$N_{\alpha,c} = O\left(N_\alpha^{\frac{1}{1+\sigma}}\right) \tag{3.4}$$

and furthermore

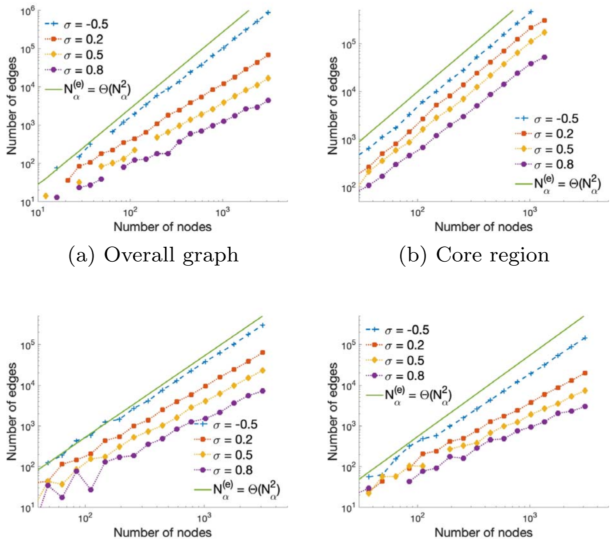
$$\frac{N_{\alpha,c}}{N_\alpha} = O\left(\alpha^{-\sigma}\right) \qquad \textit{almost surely as } \alpha \to \infty \tag{3.5}$$

We know from Equation (3.3) that when $\sigma \in (0,1)$, $N_\alpha^{(e)} = O\left(N_\alpha^\beta\right)$ in each region, where σ is the parameter of the base Lévy measure. We then have that

$$\beta = \begin{cases} \frac{2}{1+\sigma} & \text{overall} \\ 2 & \text{in the core region} \\ \frac{2}{1+\sigma} & \text{in the periphery region} \\ \frac{2}{1+\sigma/2} & \text{in the core-periphery region} \end{cases}$$

When $\sigma < 0$, $\beta = 2$ in each region.

In Figure 3, we see a graphical representation of these results. From Figure 3(a) we see that the number of edges grows quadratically with the number of nodes when $\sigma < 0$ (and thus the graph is dense) and otherwise grows with power-law exponent $\frac{2}{1+\sigma}$. We see similar behaviour for the periphery and core-periphery regions, but in Figure 3(b) we see that the core region is dense for any value of σ. In Appendix B we present some more empirical results on the effects of varying the model parameters on the degree distribution, sparsity properties and core proportion.



(a) Overall graph

(b) Core region

(c) Core-periphery region

(d) Periphery region

Fig 3. *Relationship between nodes and edges for varying σ for (a) the overall graph, (b) the core, (c) the core-periphery region and (d) the periphery. In each case we use $K = 2, \tau = 1, a = 0.2, b = \frac{1}{K}$ and vary α to generate different sized graphs. We also plot a quadratic relationship between nodes and edges, for comparison.*

4. Experiments

In this section, we present a series of experiments where we perform posterior inference using our model. We first introduce the MCMC algorithm we use to perform this inference. We then detail experiments performed on simulated and real data sets. Our goal here is twofold. Firstly, we show the ability of our model to capture both the core-periphery structure and sparsity properties of networks. Secondly, we emphasise the benefits of our approach over that of [41]. Finally, when considering our model as purely a core-periphery detection method, we show that it compares favourably against some other standard methods.

4.1. Posterior inference

In order to test our model, we design an MCMC algorithm to perform posterior inference, based on those of [6] and [41]. We assume that we have observed a set of connections $(z_{ij})_{1\leq i,j,\leq N_\alpha}$ where N_α is the number of nodes with at least one connection. We want to infer the parameters $(w_{i1}, \ldots, w_{iK})_{i=1,\ldots,N_\alpha}$. We also want to estimate the sums of the parameters for the nodes with no connection $(w_{*1}, \ldots, w_{*K})$, the hyperparameters ϕ of the mean intensity ρ and the parameter α which is also assumed to be unknown. Thus the aim is to sample from the posterior

$$p((w_{1k},\dots,w_{N_\alpha k},w_{*k})_{k=1,\dots,K},\phi,\alpha \mid (z_{ij})_{1\leq i,j,\leq N_\alpha}) \tag{4.1}$$

In order to do this, we define an algorithm which uses Metropolis-Hastings (MH) and Hamiltonian Monte Carlo (HMC) updates with a Gibbs sampler to perform posterior inference. The details of the algorithm are provided in Appendix C.

4.2. Simulated data

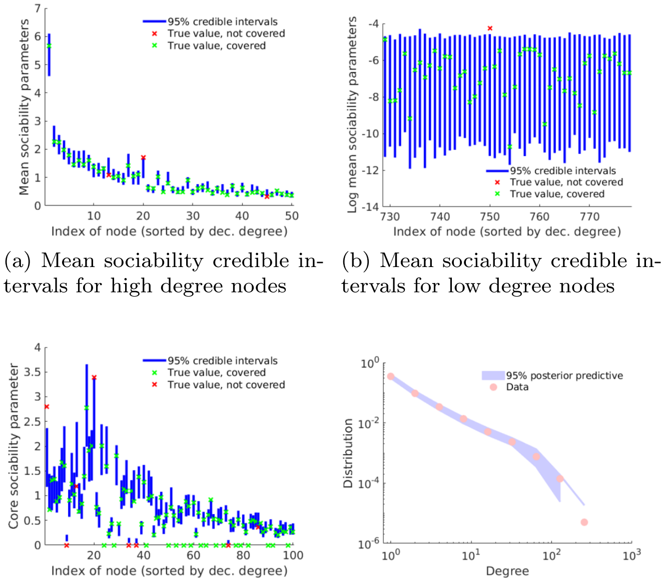
We start by considering simulated data. This is firstly in order to study the convergence of our MCMC algorithm. Secondly, it allows us to compare our model to that of [41] in a setting where have complete information about the ground truth.

In order to study the convergence of our algorithm, we generate synthetic data from the SparseCP model described in Section 2, with the construction given in 2.2. We generate a graph with $K = 2$, i.e. with core-periphery structure but no community structure. We use parameters $\alpha = 200$, $\sigma = 0.2$, $\tau = 1$, $b = \frac{1}{K}$, $a = 0.2$. In our case, the sampled graph has 778 nodes and 5984 edges.

We fit our model to the simulated network, placing vague $\Gamma(0.01, 0.01)$ priors on the unknown parameters α, $1-\sigma$, τ, a_k and b_k. We initialize the parameters with an initialization run (as described in Section C.4 of Appendix C) of 10 000 steps. We then run 3 parallel MCMC chains of 500 000, discarding the first 375 000 samples as burn in. Finally, we thin the remaining 125 000 to give a sample size of 500. Trace plots and convergence diagnostics are given in Section D.1.1 of Appendix D.

Our model accurately recovers the mean sociability parameters (in this case, the mean $\bar{w}_i = \frac{1}{2}(w_{i1} + w_{i2})$ of the core and overall sociability parameters) of both high and low degree nodes, providing reasonable credible intervals in each case as shown in Figures 4(a) and 4(b). In Figure 4(c) we see some high degree periphery nodes in particular being mis-classified into the core. However, we expect that these high degree periphery nodes will be some of the hardest to correctly classify. By generating 5000 graphs from the posterior predictive distribution, we also see that the model fits the empirical power-law degree distribution of the generated graph well, as shown in 4(d). Here, and in other plots of the posterior predictive degree distribution, the lower bound of the 95% credible interval is equal to 0 for some degree bins, however the plot is on the log scale, and so we cannot properly represent the interval in this case. Instead, only the upper bound of the interval is shown, with all the space below this line also within the credible interval.

Moreover, we see that we are able to very accurately recover the classification into core and periphery. In our generated graph, 137 out of 144 nodes are classified correctly into the core, with 631 out of 634 are classified correctly into the periphery. Importantly, the core is not simply comprised of the nodes with the highest degrees. This comes from the form of the probability of connection between two nodes; $1 - e^{-\sum_{k=1}^{K} w_{ik}w_{jk}}$. Thus a node i in the periphery, and so



(a) Mean sociability credible intervals for high degree nodes

(b) Mean sociability credible intervals for low degree nodes

(c) Core sociability (w_{i1}) credible intervals for high degree nodes

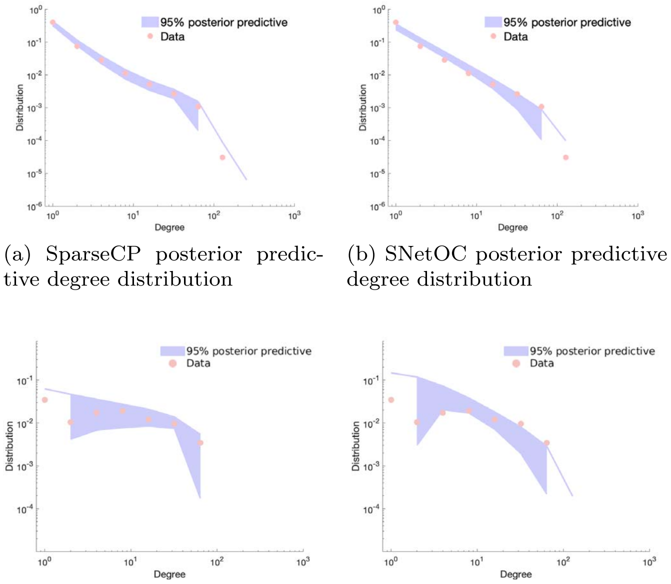
(d) Posterior predictive degree distribution

Fig 4. *Fitting the SparseCP model to data simulated from a graph generated with* $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = \frac{1}{K}, a = 0.2$. *Mean sociability* ($\frac{1}{2}(w_{i1} + w_{i2})$) *credible intervals for* (a) *the* 50 *nodes with the highest degrees,* (b) *the* 50 *nodes with the lowest degree.* (c) *Core sociability* (w_{i1}) *for the* 100 *nodes with highest degrees. In each case the true values are coloured green if they are covered by the credible interval, and red if not.* (d) *Posterior predictive degree distribution. When the lower posterior predictive interval bound is 0, we plot only the upper interval bound.*

with $w_{i1} = 0$, can have a high probability of connection with another node j if they both have large overall sociabilities w_{i2}, w_{j2}. Thus, nodes not in the core can still have a high degree. Conversely, nodes in the core but with low overall sociabilities may have low degrees. In Figure 4(c) we see the credible intervals for the core sociability parameters, and we see that there are several high degree nodes in the periphery which are identified correctly. This means that our algorithm is not simply classifying the highest degree nodes into the core.

In Appendix D, we also test our model on a graph with core-periphery and community structure. In this case, K is taken to be 4 in order to generate a graph with a core-periphery structure and 3 overlapping communities. We show that we are able to recover both the core-periphery and community structure in this case.

Comparison with the SNetOC model of [41]. Our model is an extension of that of [41], to allow the modelling of core-periphery structure as well as community structure. The question arises as to whether their SNetOC model could



(a) SparseCP posterior predictive degree distribution

(b) SNetOC posterior predictive degree distribution

(c) SparseCP posterior predictive degree distribution for core subgraph

(d) SNetOC posterior predictive degree distribution for core subgraph

FIG 5. *Posterior predictive degree distributions for a graph simulated from the SparseCP model with parameters $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = 1, a = 0.2$. The top row gives the posterior predictive degree distributions using (a) the SparseCP model and (b) the SNetOC model. The bottom row gives the posterior predictive degree distributions for the subgraph formed by the nodes which are in the true core. This is calculated using the nodes estimated to have (a) $w_{i1} > 0$ for the SparseCP model, (b) $w_{i1} > 0.014$ for the SNetOC model. When the lower posterior predictive interval bound is 0, we plot only the upper interval bound.*

perform the same role as ours, with the core thought of as another community. Indeed, when examining real world networks we will study an example in which [41] identify a *Hub* community which is similar to the core identified by our model.

Here, we show some of the problems that can occur when trying to use the SNetOC model for core-periphery detection, and the advantages that our model can provide. We start by generating a graph from the SparseCP model with the parameters $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = 1, a = 0.2$. This choice of parameters leads to a larger relative core size than in our previous example. We then fit both the SparseCP model and the SNetOC model to this, and compare the results. We do this in two ways, firstly the fit of the posterior predictive degree distribution to the empirical degree distribution, and secondly the classification accuracy of the estimated core.

In Figure 5(a) we see the posterior predictive degree distribution for the SparseCP model, with the corresponding plot for the SNetOC model given in 5(b). To compare these distributions, we can measure the distance between the

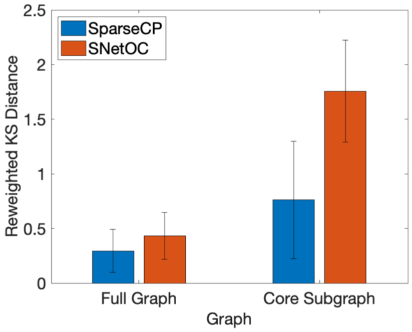


Fig 6. *Reweighted Kolmogorov Smirnov distance for assessing the degree distribution for the network generated from the SparseCP model with parameters* $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = 1, a = 0.2$.

posterior predictive degree distributions and the empirical degree distribution. It is well known that the Kolmogorov Smirnov (KS) statistic has poor sensitivity to deviations from the empirical distribution that occur in the tails [30]. So, we give here the reweighted KS statistic suggested by [8], which weights the tails more strongly:

$$D = \max_{x \geq x_{min}} \frac{|S(x) - P(x)|}{\sqrt{P(x)(1 - P(x))}} \tag{4.2}$$

where $S(x)$ is the CDF of observed degrees, $P(x)$ is the CDF of degrees of graphs sampled from the posterior predictive distribution and x_{min} is the minimum value among the observed and sampled degrees.

We see from the left hand side of Figure 6 that our model gives a better fit in terms of the weighted KS distance. Moreover, the SparseCP model (correctly) estimates σ to be positive, whilst the SNetOC model estimates σ to be negative. This shows one important benefit of our model, namely that it can model sparse networks with a large core well (the true core proportion in this case is around 0.3). Conversely, we see here an example of where the SNetOC model fails to differentiate between a dense network and a sparse network with a dense subgraph. It is important to note that in some situations, the SNetOC model correctly models the empirical degree distribution. One such example is the graph that we use to test the convergence of our MCMC sampler, in which the core is significantly smaller.

We can also compare the two models in terms of the core they identify. In the SparseCP model the binary variable $\beta_{i1} = \mathbb{1}_{w_{i1}>0}$ indicates if the node i is in the core or not. We can then report the posterior distribution of this variable, or the point estimate $\widehat{\beta}_{i1} = \arg\max_{k=0,1} \Pr(\beta_{i1} = k \mid \text{data})$.

For the SNetOC model it is more difficult to perform this classification. A naive way to proceed is to assign each node to the community with the highest weight, and then identify the core as the subgraph of nodes with the highest density. However, in the SNetOC model nodes can belong to multiple communities, and a node can belong to the core, and still have a higher weight for another community. This method of classification gives poor results.

To provide a fairer comparison, we can also perform classification with the SNetOC model by assigning node i to the core if $w_{ic} > T$, where c denotes the index of the hub community, and $T > 0$ is some threshold. The problem here is that there is no objective way to choose T, without resorting to some cross-validation. In this simulated example, we can choose the threshold so that the number of nodes that are placed into the hub community by this classification method is equal to the number that are in the true core. This corresponds to a threshold of $T = 0.014$ in this case. If we do this, the classification accuracies of the SparseCP and SNetOC methods are very similar, and very close to 1. However, this method is not applicable for real data where we do not know the size of the true core in advance.

Even in this "best-case" scenario for the SNetOC model where we classify based on thresholding the w_{ic} optimally in some sense, we can still see the advantage of the SparseCP model by looking at the fit of the posterior predictive to the empirical degree distribution of the subgraph formed by the nodes in the (true) core. In Figures 5(c) and 5(d) we see that the SparseCP model is performing significantly better than the SNetOC model in this regard. This is confirmed by the weighted KS distances that we see in the right hand side of Figure 6.

When comparing the SparseCP and SNetOC model on real data sets, we will see many of the same features that we have noted here. In particular, the problems that the SNetOC model has with modelling a sparse graph with a relatively large dense core, and the problems with choosing a thresholding level to perform classification.

4.3. Real world networks

We now consider the setting of real data. There are many classical examples of networks with core-periphery structure. Our model is designed to detect this structure in sparse networks with a power-law degree distribution. We apply our method on two real world networks, one that has a power-law distribution and one that does not. The proposed method performs well in the power-law setting but also produces reasonable and interpretable results in the non-power-law case. The power-law network we consider is the **USairport** network of airports with at least one connection to a US airport in 2010[2]. This network was previously considered by [41], and we compare the two methods to see the benefits of using a core-periphery model in this case. The other network we consider is the **Trade** network of historical international trade[3] for which the core-periphery structure has been previously studied [12, 14]. In Table 1, we give the size of these networks, the value of K, the estimated relative size of the core and the estimated value of σ.

[2] `http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=292`

[3] `https://web.stanford.edu/~jacksonm/Data.html`

For the **USairport** data set we set $K = 4$. This follows from the work of [41], who found that there were four interpretable communities in the network. Three of these corresponded to geographical regions, with the final one corresponding to *Hub* airports. We therefore take the same choice of K to see if our model can identify the same interpretable structure, and to compare the two methods. For the **Trade** network we take $K = 2$, as in this case we do not concern ourselves with community structure, but simply look to see if it is possible to identify core-periphery structure even when the network does note have a power-law degree distribution. In each case, we assume vague $\Gamma(0.01, 0.01)$ priors on the unknown parameters α, $1 - \sigma$, τ, a_k and b_k.

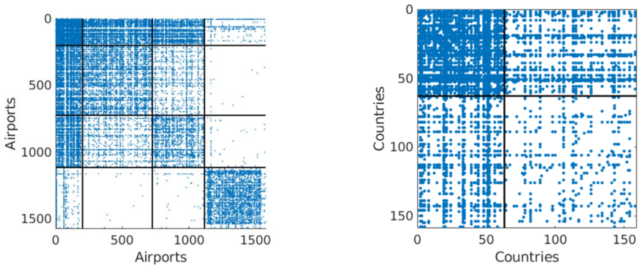
Table 1
Real World Networks and Summary Statistics.

Name	Nb Nodes	Nb Edges	K	Est Core Prop	$\widehat{\sigma}$
USairport	1574	17215	4	0.13	0.22
Trade	158	1897	2	0.39	-0.77

4.3.1. US Airport network

The first real world network we consider is the **USairport** network of airports with at least one connection to a US airport in 2010. Airport networks such as this have been seen to have a core-periphery structure [21, 12]. Furthermore, one of the communities identified by [41] are the *Hub* airports, highly connected airports with no preferred location. [39] suggest that this network could more accurately be modelled by a core-periphery model. When comparing the results of our model to that of [41], we find that this is indeed the case. In order to compare both models, we take $K = 4$ when fitting the SparseCP model to the network, to retain the three other communities identified using the SNetOC model.

We run 3 parallel MCMC chains with an initialization run of 10 000 iterations followed by 10 000 000 iterations. We then discard the first 5 000 000 samples as burn in and thin the remaining 5 000 000 to give a sample size of 500. Trace plots and convergence diagnostics are reported in Section D.2.1 of Appendix D. Note that despite the large number of iterations, it seems that the sampler has not fully converged yet. Nonetheless, we observed that increasing the number of iterations does not the change significantly the values of the core-periphery parameters of interest and the overall interpretation of the network. In Figure 7(a) is given the adjacency matrix formed by ordering the nodes firstly by core and periphery, and then by their highest community sociability. From this, we can see the core-periphery and community structure identified by our model. The three communities have the same geographical interpretation as in [41], corresponding to *East*, *West* and *Alaska*. In Figure 8, we report the weights of a selection of nodes in the core and in the periphery. Periphery airports with large degree are generally large regional airports, with many connections within the *East* or *West* communities. The final community corresponds to Alaskan airports, and we can see from Figure 7(a) that airports in this community generally do not have many connections to the core.



(a) US Airport network

(b) World Trade network

Fig 7. *Adjacency matrices for the* (*a*) *US Airport network and* (*b*) *World Trade network. In each case the nodes are ordered into core and periphery, as estimated using the SparseCP model. For the US Airport network, the nodes are further ordered by highest community weight.*

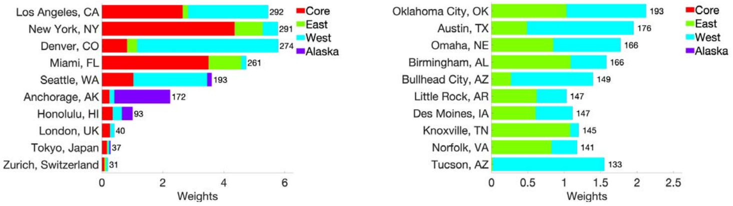
If we investigate the classification into core and periphery in more detail, we see several interesting features. Firstly, when looking at the airports placed in the core, we can compare these to the list of hub airports for various airline companies[4]. We see that 38 out of 47 of these airline hub airports are placed in the core, with some interesting exceptions. Three of the airline hub airports not placed in the core are only hubs for the parcel delivery services Fedex and UPS, whilst one is only a hub for a small charter airline. The most interesting exceptions are Chicago Midway and Dallas Love Field airports, both of which are focus cities for major airline company Southwest Airlines but are not placed in the core. A possible explanation for this is that Southwest Airlines does not operate using a traditional spoke-hub model, instead utilising a point-to-point system. In this system, transport routes are directly between locations, rather than via central hubs. This means that we do not see the presence of core airports with large numbers of connections to airports in many different locations. Connections in this network can be better explained by geographical proximity, and the focus cities are not necessarily core nodes in the sense of our model.

Secondly, we see that the SparseCP model does not simply classify by degree, the airports that we see in Figure 8(b) have degrees higher than many core nodes. As described in the simulated data study, this comes from the form of the probability of connection between two nodes, which can be high even if neither is in core.

If we look at the core nodes with low degrees, we can see a clear pattern here. Most of these airports are major international hubs such as in London Heathrow, Frankfurt International, Zurich and Tokyo Narita. Another airport that appears here is Honolulu International. In each case, it is not surprising that these airports are placed in the core, as most of the connections are long-distance flights to the major US hub airports.

Investigating the overestimation of high degree nodes more carefully, we see that for our model and that of [41], the posterior distribution on b_4, the parame-

[4] https://en.wikipedia.org/wiki/List_of_hub_airports



(a) Selection of airports in the core

(b) Periphery airports with highest degrees

Fig 8. *Values of the weights for* (*a*) *a selection of core nodes and* (*b*) *the periphery nodes with the highest degrees. The coloured bars indicate the respective weights for the core and each community. The numbers give the degrees of the nodes.*

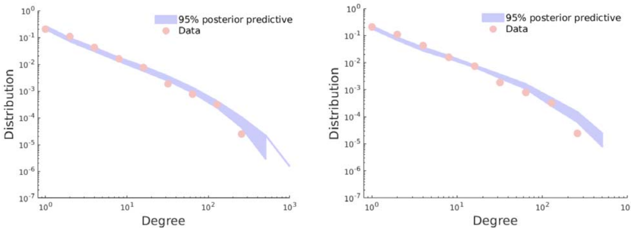
ter entering the gamma distribution F corresponding to the Alaskan community (the 3rd community identified by our model) concentrates on values very close to 0 (see Figure 33 in Appendix D). This increases the posterior variance of the sociabilities w_{i4}, and leads to some graphs with very high degree nodes being generated when simulating from the posterior predictive distribution. A possible reason for this is that the Alaskan airports exhibit a different type of behaviour, not well explained by our model. Alaskan airports are densely connected to each other but not to the other communities, or the core nodes. Conversely, the *East* and *West* communities contain large numbers of core nodes, and also have more connections between communities.

In order to investigate this possible source of model misspecification further, we repeat our analysis of this network, taking out all of the Alaskan airports, and any airport which only has connections to Alaskan airports. In Appendix E we present the results. We see that the overestimation problem is no longer present, but overall the fit to the degree distribution is no better than for the full model. Furthermore, if we examine the nodes that are placed into the core and periphery, there is not very much difference.

Comparison with the SNetOC model of [41]. We now compare the fit of our model and the SNetOC model to the **USairport** network. In Figures 9(a) and 9(b), we report the 95% posterior predictive intervals for the degree distribution for both models. The SparseCP provides a slightly better fit to the degree distribution, in particular for large degree nodes.

In Figure 10(a) we report some statistics to measure the distances between the posterior predictive degree distributions from each model and the empirical degree distribution. Firstly, we see that our model is performing significantly better in terms of the reweighted KS distance defined in Equation (4.2). To identify in which tail the deviation between the two models occurs, we also report the Rényi statistics

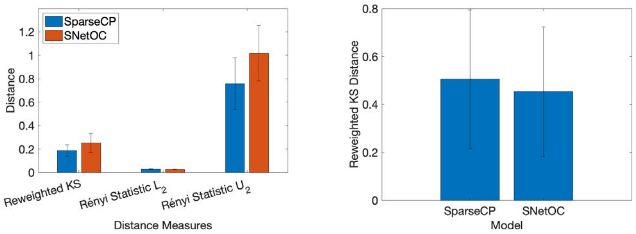
$$L_2 = \sup_{x \geq x_{min}} \frac{P(x)}{S(x)}, \quad U_2 = \sup_{x \leq x_{max}} \frac{1 - P(x)}{1 - S(x)} \tag{4.3}$$



(a) SparseCP posterior predictive degree distribution

(b) SNetOC posterior predictive degree distribution

Fig 9. *Posterior predictive degree distributions for the US Airport network, using* (*a*) *the SparseCP model and* (*b*) *the SNetOC model. When the lower posterior predictive interval bound is 0, we plot only the upper interval bound.*



(a) US Airport network

(b) World Trade Network

Fig 10. *Reweighted Kolmogorov Smirnov distance and Rényi statistics for assessing the degree distribution for the* (*a*) *US Airport network and* (*b*) *World Trade Network.*

where $S(x)$ is the cdf of observed degrees, $P(x)$ is the posterior predictive cdf of the degrees and x_{min} and x_{max} are respectively the minimum and maximum observed degrees. Such statistics have been used as an alternative to the standard KS statistics, to measure departure in the tail [30]. L_2 is more sensitive to a departure for the low degrees, and U_2 for the high degrees. Both models perform similarly well for low degrees, but the fit of the SparseCP model is better for the large degree nodes.

We now compare the classification accuracy of the two models. We use as ground truth for the core nodes the 47 US airports designated as hubs by airlines.

For the SparseCP model, for $x \in [0,1]$ let $\widehat{\beta}_{i1}(x) = 1$ if $\Pr(w_{i1} > 0 \mid data) \geq x$ and 0 otherwise. For the SNetOC model, let $\widehat{\beta}_{i1}(x) = 1$ if $\Pr(w_{i1} > T \mid data) \geq x$ and 0 otherwise, for some threshold $T > 0$. For comparison with SNetOC, we also compute the same estimate under our model for various thresholds T (called SparseCP variable threshold). For each method and for a range of thresholds T, we plot the true positive rate vs the false positive rate for $x \in [0,1]$, and calculate the Area Under the Curve (AUC), as a measure on the classification accuracy. We also compare against the naive classification method for the SNetOC model mentioned in the previous section, where a node is assigned to the

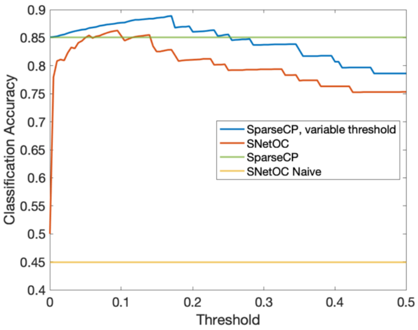


Fig 11. *Classification accuracy of the airline hubs in the US Airport network for the SparseCP and SNetOC models with varying threshold.*

hub community if its hub community weight is the highest amongst all of its community weights. We call this method SNetOC Naive.

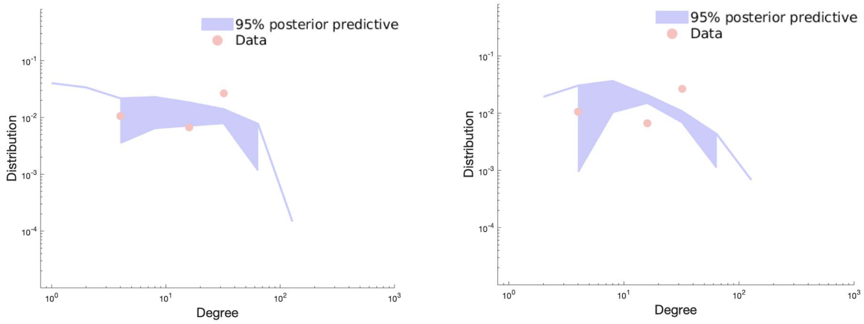
Figure 11 compares the classification accuracy from the SparseCP model (in green) to that obtained from the SNetOC model with different thresholds (in red). We see that with the right choice of threshold, the SNetOC model can slightly outperform ours, but for other thresholds it performs far worse. We also plot (in blue) the AUC obtained from our model if we do not threshold $\hat{w}_{i1}$ at 0, but instead at higher values. This can be thought of as performing classification using a different loss function, which could be useful in a setting where we are more concerned with misclassifying periphery nodes as core than the reverse. For any value of T, the proposed model provides a higher accuracy than the SNetOC model. Moreover, in practice, we do not have an objective way to choose a particular threshold (without resorting to cross-validation). The naive SNetOC classification does not require a choice of threshold, however we see here that it provides a far lower classification accuracy. This suggests that our model provides a better way to identify core-periphery structure.

Furthermore, if we plot the degree distribution of the subgraph formed by the US airport hubs, we see that it is very far from power-law. In Figure 12 we see that the SparseCP model is capturing this degree distribution better than the SNetOC model.

4.3.2. World trade network

The next network we consider is the dyadic trade network between countries[5]. The original data details the flow of trade between pairs of countries from 1870 to 2009. In our case, we consider the single simple, undirected network found by aggregating the data over all the years. However, as shown by [12], when the trade data set is considered as an unweighted network, with a link between countries if there was any flow of trade between them, then the core-periphery structure

[5] https://web.stanford.edu/~jacksonm/Data.html



(a) SparseCP posterior degree for core subgraph

(b) SNetOC posterior degree for core subgraph

Fig 12. *Posterior predictive degree distributions for the nodes of the US Airport network in the "true" core subgraph, as defined by the airline hubs. When the lower posterior predictive interval bound is 0, we plot only the upper interval bound.*

is quite weak. There is a strong core-periphery structure if the weighted (by volume of trade) network is considered. As our method currently cannot deal with weighted networks, we instead use a cutoff method, forming a binary network by only considering trade links over a certain volume. The cutoff was chosen to give a network with a heterogeneous degree distribution which was close to a power-law. With a very low cutoff, the network is quite dense, with most countries connected to each other. With a high cutoff, only a small number of large countries have any connections. However, we find that the degree distribution is not power-law, with σ estimated to be negative. We found that this was the case even with different values of the cutoff.

In order to perform posterior inference, we run 3 MCMC chains with an initialization run of 10 000 steps and full chain lengths of 200 000. Trace plots and convergence diagnostics are reported in Section D.2.2 of Appendix D, suggesting the convergence of the MCMC sampler.

As we see in Table 1, σ is estimated to be negative in this case. This means that this network does not fit into our definition of a sparse graph as defined in Definition 2.2. Nevertheless, in Figure 13 we plot the posterior predictive distribution and we see that we can estimate the degree distribution fairly well, albeit with a large posterior predictive interval. Furthermore, we will see that we can obtain an interpretable classification of countries into core and periphery. Therefore, our model is still producing useful results despite the network not technically fitting into our framework.

As with the US Airport network, we calculate the reweighted KS distances in Figure 10(b). We again compare against the model of [41], with two communities. Here we see that both models perform similarly well. However, the advantage of our model is that it gives a discrete classification between core and periphery. we see this structure clearly from the adjacency matrix in Figure 7(b).

In Figure 14 we see the world map, coloured by the value of mean sociability

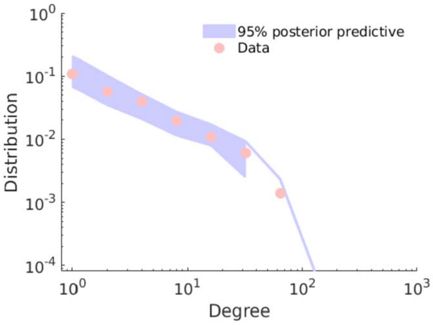


Fig 13. *Posterior predictive degree distribution for the World Trade Network, using the SparseCP model. When the lower posterior predictive interval bound is 0, we plot only the upper interval bound.*

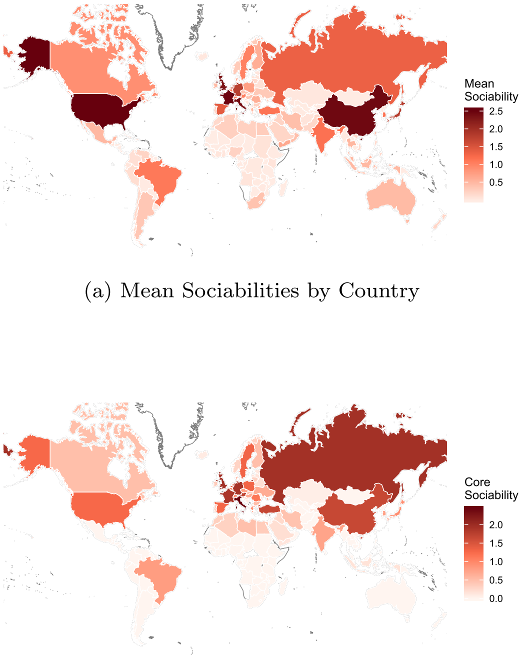
parameter in Figure 14(a), and by the value of the core sociability parameter in Figure 14(b). We see that the core consists largely of the large, developed countries that we expect, as well as some smaller European countries. This fits with other results that have been obtained in the literature [12]. Comparing the two plots, some of the more interesting results are countries that have a relatively high core sociability compared to their mean sociability. These include countries such as Russia, and indicate countries that trade predominantly with other countries in the core. Conversely, we see other countries such as the US and China which have relatively lower core sociabilities. These are countries which trade more internationally, with countries in the core and periphery.

4.4. Comparisons

Finally, we want to compare our method against other standard methods of core-periphery detection. In order to do this we generate simulated core-periphery networks, and run each algorithm on them. We can then calculate the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve [18] in order to determine the accuracy of the classification of each algorithm. In this case, we simply compare the binary core-periphery classification of each method, but we can still use the AUC as a measure of the accuracy of this binary classification. The methods that we compare against are: the algorithm of [4], the MINRES algorithm of [5] and [28], the Stochastic Block-Model (SBM) of [44], the three different methods of [10], the aggregate core score method of [39] and the method of [11]. In each case, we implement the methods using the software found in the `cpalgorithm` Python package[6]. While the method of [39] gives continuous coreness parameters, we convert these to binary classifications in the same way that they do in their simulation study.

As we have previously noted, our focus here is on model-based approaches. The only other model-based approach amongst these is that of [44]. Our ap-

[6] https://core-periphery-detection-in-networks.readthedocs.io/en/latest/



(a) Mean Sociabilities by Country

(b) Core Sociabilities by Country

Fig 14. *Core Periphery Structure of the World Trade network, as estimated by the SparseCP model.* (*a*) *gives the mean sociabilities* $\frac{1}{2}(w_{i1} + w_{i2})$ *for each country,* (*b*) *gives the core sociabilities* w_{i1}.

proach already provides several advantages over the rest of the models considered, such as the ability to model the degree distribution of the networks being studied. However, here we focus on the classification accuracy and how it compares to some classic and more contemporary alternatives. Specifically, we are interested in comparing the accuracy in the setting for which our model is designed: the modelling of sparse graphs with a core-periphery structure and power-law degree distribution.

4.4.1. Comparison on in-model simulated data

The first comparison we do is using the SparseCP model to generate power-law, core-periphery networks. Of course, we expect our model to perform very well here, and indeed we see this as we compare the AUC for each model. We vary the strength of the core-periphery structure by varying b in our model, which varies the relative sociabilities of the nodes in the core and periphery. We run 20 simulations for each value of b, and adjusting the value of α to keep the number of nodes roughly equal in each case. We then measure the average classification accuracy in each case. In Figure 15 we present the results for varying val of b.

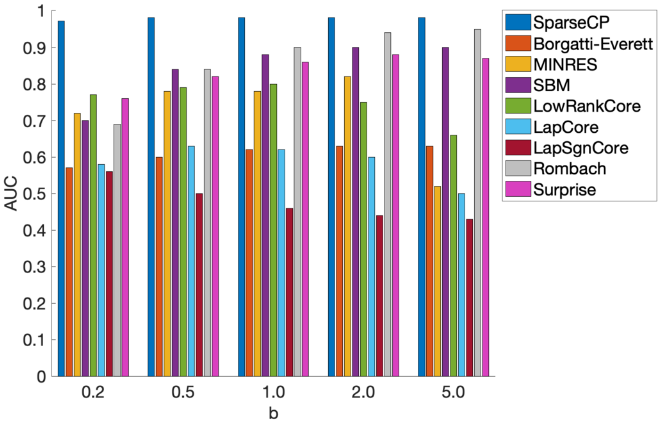


Fig 15. *AUC of different classification methods for the graph generated from the SparseCP model with parameters $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, a = 0.2$, and differing values of b.*

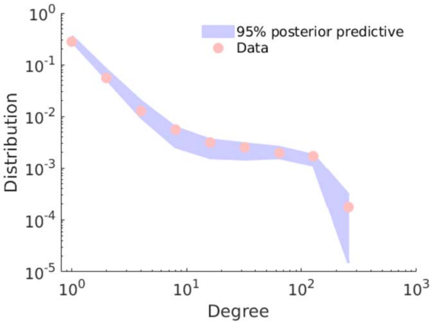


Fig 16. *Posterior predictive degree distribution using the SparseCP model for the graph generated from the SparseCP model with parameters $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = 5, a = 0.2$.*

As b increases, the average relative size of the core increases from 0.12 to 0.57, whilst the average number of nodes stays roughly constant. We can see that classifying core and periphery generally becomes easier to do as b is increased. However, we see that our method achieves the highest accuracy in each case.

The case $b = 5$ is an extreme case, with the core comprising almost 60% of the network, and in this case some of the other methods perform badly, with some methods placing all the nodes in the core in some of the simulations. Recalling our definition of core-periphery networks, we see from Figure 16 that the degree distribution is very far from a power-law, due to the large number of core nodes. Nevertheless, it is encouraging to see that in this case we are still able to accurately recover the degree distribution.

There are a few reasons why we might expect to see this difference in classification performance. The only other model-based approach, based on the SBM [44], cannot account for power-law degree distributions and tends to classify nodes according to their degree. Other non-model-based approaches have similar issues.

In an attempt to allow a fairer comparison, we also test our method against the alternatives mentioned above on another simulated network, again with a power-law degree distribution, but in this case not generated using our model.

4.4.2. Comparison on out-of-model simulated data

The second comparison we do is using the simulated core-periphery networks of [21]. Our algorithm is based on theirs, but differs slightly to allow us to tune how well the core and periphery can be separated by degree. We construct networks as follows:

1. Generate degrees $m_1 < m_2 < \ldots < m_N$ from a power-law distribution. These are the desired degrees and can be thought of as stubs, as in the configuration model [33].
2. Place node i in the core with probability q_i, where q_i is given by

$$q_i = \frac{1}{1 + \exp\left(-2\kappa(m_i - m_{min})\right)}$$

   where κ and m_{min} are parameters that we use to tune the model. Call the set of core nodes $\mathcal{C}_\mathcal{H}$ and the set of periphery nodes $\mathcal{P}_\mathcal{H}$.
3. Go through each of the nodes $i \in \mathcal{C}_\mathcal{H}$ in increasing order of degree (node i has degree m_i) and for each node i attach its stubs to those of nodes j with $m_j \geq m_i$ as long as the degree of j is less than m_j.
4. Attach the remaining stubs randomly and make them into edges if they do not form loops or multiple edges.

The form of q_i is a standard approximation to a Heaviside step function. When κ is large, the function approximates a step function with a jump at m_{min}. In this case, the core is comprised of only the nodes with degrees $m_i \geq m_{min}$. However, for smaller κ we have low degree nodes entering the core and high degree nodes entering the periphery. In Figure 17 we compare our method to the alternatives for varying κ As in our previous simulation study, we calculate the area under the ROC curve, averaged over 20 realisations of networks for each value of κ.

We see that when κ is large, and the core and periphery are essentially divided by degree, our method performs very well, but so do those of [44] and [39]. As we decrease κ, the other methods fail to perform as well, while ours retains a high accuracy. Therefore, we see that our method also outperforms the alternatives when applied to simulated networks that are not generated using our model. Specifically, we do better especially when the core and periphery nodes are not split simply by degree. We have seen in the **USairport** example that the form of the probability of connection between nodes is what allows nodes in the periphery to have high degree, and vice-versa. This explains why our model performs well here.

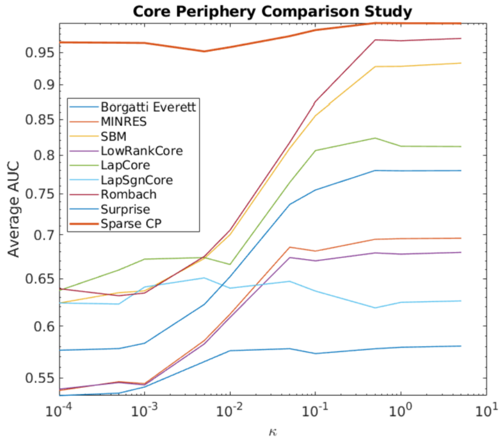


Fig 17. *Average classification accuracy (as measured by the AUC) for a range of standard methods for core-periphery detection, along with classification accuracy using the SparseCP model. The accuracies are calculated for different values of the parameter κ, which controls the extent to which core and periphery can be divided by degree.*

We must also note here that some of the methods we compare against here are very fast, taking only seconds to produce a classification into core and periphery. However, our sampler generally takes ~ 5 minutes to run on a standard desktop computer. Furthermore, as discussed earlier in our case the aim is not just to perform classification, but to estimate the parameters of a generative model.

5. Conclusions

In this work, we provide a precise definition of a sparse network with a core-periphery structure, based on the sparsity properties of the subgraphs of core and periphery nodes. Building on earlier work from [6, 41], we then present a class of sparse graphs with such properties. We obtain theoretical results on the sparsity properties of our model. Specifically, we see that the SparseCP model generates a core region which is dense, and a periphery region which is sparse. Theoretical results on the relative size of the core are also obtained.

We provide methods to simulate from this class of graphs, and to perform posterior inference with this class of models. We demonstrate that our approach can detect interpretable core-periphery structure in two real-world airport and trade networks, while providing a good fit to global structural properties of the networks. When restricting ourselves to simply looking at core-periphery classification accuracy, we see that it compares favourably against various alternatives when tested in the power-law setting.

A property of the SparseCP model is that the relative size of the core tends to zero as the size of the graph increases. In some applications, we may instead want to have a network which is overall dense, but with a sparse periphery region, where the relative size of the core is bounded above zero. Furthermore, whilst our model can currently accommodate the existence of multiple communities as well as a core-periphery structure, it currently cannot be used to model networks with multiple core-periphery pairs. Work has been done in the literature on the detection of multiple such pairs [25] and it could be valuable to extend our model to this setting.

References

[1] Ayed, F., Lee, J. and Caron, F. (2019). Beyond the Chinese restaurant and Pitman-Yor processes: statistical models with double power-law behavior. In *International Conference on Machine Learning (ICML 2019).*

[2] Barabási, A.-L. et al. (2016). *Network science.* Cambridge university press.

[3] Bingham, N. H., Goldie, C. M. and Teugels, J. L. (1989). *Regular variation* **27**. Cambridge university press. MR0898871

[4] Borgatti, S. P. and Everett, M. G. (2000). Models of core/periphery structures. *Social networks* **21** 375–395.

[5] Boyd, J. P., Fitzgerald, W. J., Mahutga, M. C. and Smith, D. A. (2010). Computing continuous core/periphery structures for social relations data with MINRES/SVD. *Social Networks* **32** 125–137.

[6] Caron, F. and Fox, E. B. (2017). Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79** 1295–1366. MR3731666

[7] Caron, F., Teh, Y. W., Murphy, T. B. et al. (2014). Bayesian nonparametric Plackett–Luce models for the analysis of preferences for college degree programmes. *The Annals of Applied Statistics* **8** 1145–1181. MR3262549

[8] Clauset, A., Shalizi, C. R. and Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review* **51** 661–703. MR2563829

[9] Csermely, P., London, A., Wu, L.-Y. and Uzzi, B. (2013). Structure and dynamics of core/periphery networks. *Journal of Complex Networks* **1** 93–123.

[10] Cucuringu, M., Rombach, P., Lee, S. H. and Porter, M. A. (2016). Detection of core–periphery structure in networks using spectral methods and geodesic paths. *European Journal of Applied Mathematics* **27** 846–887. MR3570008

[11] de Jeude, J. v. L., Caldarelli, G. and Squartini, T. (2019). Detecting core-periphery structures by surprise. *EPL (Europhysics Letters)* **125** 68001.

[12] Della Rossa, F., Dercole, F. and Piccardi, C. (2013). Profiling core-periphery network structure by random walkers. *Scientific reports* **3** 1467.

[13] Erdös, P. (1959). On random graphs. *Publicationes mathematicae* **6** 290–297.

[14] Fagiolo, G., Reyes, J. and Schiavo, S. (2010). The evolution of the world trade web: a weighted-network analysis. *Journal of Evolutionary Economics* **20** 479–514. MR2586881

[15] Fricke, D. and Lux, T. (2015). Core–periphery structure in the overnight money market: evidence from the e-mid trading platform. *Computational Economics* **45** 359–395.

[16] Gelman, A., Rubin, D. B. et al. (1992). Inference from iterative simulation using multiple sequences. *Statistical science* **7** 457–472. MR1294072

[17] Griffin, J. E. and Leisen, F. (2017). Compound random measures and their use in Bayesian non-parametrics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79** 525–545. MR3611758

[18] Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **143** 29–36.

[19] Herlau, T., Schmidt, M. N. and Mørup, M. (2016). Completely random measures for modelling block-structured sparse networks. In *Advances in Neural Information Processing Systems 29 (NIPS 2016).*

[20] Holland, P. W., Laskey, K. B. and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks* **5** 109–137. MR0718088

[21] Holme, P. (2005). Core-periphery organization of complex networks. *Physical Review E* **72** 046111.

[22] Janson, S. (2011). Probability asymptotics: notes on notation. *arXiv preprint 1108.3924*.

[23] Kingman, J. (1967). Completely random measures. *Pacific Journal of Mathematics* **21** 59–78. MR0210185

[24] Kingman, J. F. C. (1993). *Poisson Processes. Oxford science publications.* Clarendon Press. MR1207584

[25] Kojaku, S. and Masuda, N. (2017). Finding multiple core-periphery pairs in networks. *Physical Review E* **96** 052313.

[26] Lijoi, A., Mena, R. H. and Prünster, I. (2007). Controlling the reinforcement in Bayesian non-parametric mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69** 715–740. MR2370077

[27] Lijoi, A., Prünster, I. et al. (2010). Models beyond the Dirichlet process. *Bayesian nonparametrics* **28** 342. MR2730661

[28] Lip, S. Z. (2011). A fast algorithm for the discrete core/periphery bipartitioning problem. *arXiv preprint 1102.5511*.

[29] Luo, F., Li, B., Wan, X.-F. and Scheuermann, R. H. (2009). Core and periphery structures in protein interaction networks. In *Bmc Bioinformatics* **10** S8. BioMed Central.

[30] Mason, D. M. and Schuenemeyer, J. H. (1983). A modified Kolmogorov-Smirnov test sensitive to tail alternatives. *The Annals of Statistics* 933–946. MR0707943

[31] Naulet, Z., Sharma, E., Veitch, V. and Roy, D. M. (2017). An estimator for the tail-index of graphex processes. *arXiv preprint 1712.01745*.

[32] Nemeth, R. J. and Smith, D. A. (1985). International trade and world-

system structure: A multiple network analysis. *Review (Fernand Braudel Center)* **8** 517–560.

[33] Newman, M. (2010). *Networks: an introduction.* Oxford university press. MR2676073

[34] Newman, M. E. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary physics* **46** 323–351.

[35] Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association* **96** 1077–1087. MR1947255

[36] Orbanz, P. and Roy, D. M. (2015). Bayesian Models of Graphs, Arrays and Other Exchangeable Random Structures. *IEEE Trans. Pattern Anal. Mach. Intelligence (PAMI)* **37** 437-461.

[37] Plackett, R. (1953). The truncated Poisson distribution. *Biometrics* **9** 485–488. MR0059517

[38] Regazzini, E., Lijoi, A., Prünster, I. et al. (2003). Distributional results for means of normalized random measures with independent increments. *The Annals of Statistics* **31** 560–585. MR1983542

[39] Rombach, P., Porter, M. A., Fowler, J. H. and Mucha, P. J. (2017). Core-periphery structure in networks (revisited). *SIAM Review* **59** 619–646. MR3683683

[40] Snijders, T. A. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification* **14** 75–100. MR1449742

[41] Todeschini, A., X., M. and Caron, F. (2020). Exchangeable Random Measures for Sparse and Modular Graphs with Overlapping Communities. *Journal of the Royal Statistical Society B.* MR4084173

[42] White, H. C., Boorman, S. A. and Breiger, R. L. (1976). Social structure from multiple networks. I. Blockmodels of roles and positions. *American journal of sociology* **81** 730–780.

[43] Williamson, S. and Tec, M. (2018). Random clique covers for graphs with local density and global sparsity. *arXiv preprint 1810.06738*.

[44] Zhang, X., Martin, T. and Newman, M. E. (2015). Identification of core-periphery structure in networks. *Physical Review E* **91** 032803.

Appendix A: Proofs

A.1. Asymptotic notation

We first describe some further asymptotic notation used in the proof. As before we follow the notation of [22], where if $X = (X_\alpha)_{\alpha \geq 0}$ and $Y = (Y_\alpha)_{\alpha \geq 0}$ are two stochastic processes defined on the same probability space with $X_\alpha, Y_\alpha \to \infty$ a.s. as $\alpha \to \infty$, we have

$$X_\alpha = O(Y_\alpha) \text{ a.s.} \iff \limsup_{\alpha \to \infty} \frac{X_\alpha}{Y_\alpha} < \infty \text{ a.s.}$$

$$
\begin{aligned}
X_\alpha = o(Y_\alpha) \text{ a.s. } &\iff \limsup_{\alpha\to\infty} \frac{X_\alpha}{Y_\alpha} = 0 \text{ a.s.}\\
X_\alpha = \Omega(Y_\alpha) \text{ a.s. } &\iff Y_\alpha = O(X_\alpha) \text{ a.s.}\\
X_\alpha = \omega(Y_\alpha) \text{ a.s. } &\iff Y_\alpha = o(X_\alpha) \text{ a.s.}\\
X_\alpha = \Theta(Y_\alpha) \text{ a.s. } &\iff X_\alpha \asymp Y_\alpha \text{ a.s. } \iff X_\alpha = O(Y_\alpha) \text{ and } Y_\alpha = O(X_\alpha) \text{ a.s.}
\end{aligned}
$$

A.2. Proof of theorems 3.1, 3.2 and corollary 3.2.1

Let

$$
\begin{aligned}
\rho_c(dw_1, dw_2, \dots, dw_K) = \rho(dw_1, dw_2, \dots, dw_K)\mathbb{1}_{w_1>0}\\
\rho_p(dw_1, dw_2, \dots, dw_K) = \rho(dw_1, dw_2, \dots, dw_K)\mathbb{1}_{w_1=0}
\end{aligned}
$$

A straightforward adaptation of Theorem 1 in [6] and Proposition 1 in [31] yields

$$
\begin{aligned}
N_\alpha^{(e)} &\sim \frac{\alpha^2}{2} \int \left(1 - e^{-2\sum_{k=1}^K w_k w_k'}\right) \rho(dw_1, dw_2, \dots, dw_K)\rho(dw_1', dw_2', \dots, dw_K')\\
N_{\alpha,c-c}^{(e)} &\sim \frac{\alpha^2}{2} \int \left(1 - e^{-2\sum_{k=1}^K w_k w_k'}\right) \rho_c(dw_1, dw_2, \dots, dw_K)\rho_c(dw_1', dw_2', \dots, dw_K')\\
N_{\alpha,p-p}^{(e)} &\sim \frac{\alpha^2}{2} \int \left(1 - e^{-2\sum_{k=1}^K w_k w_k'}\right) \rho_p(dw_1, dw_2, \dots, dw_K)\rho_p(dw_1', dw_2', \dots, dw_K')\\
N_{\alpha,c-p}^{(e)} &\sim \alpha^2 \int \left(1 - e^{-2\sum_{k=1}^K w_k w_k'}\right) \rho_c(dw_1, dw_2, \dots, dw_K)\rho_p(dw_1', dw_2', \dots, dw_K')
\end{aligned}
$$

almost surely as α tends to infinity. It follows that

$$
N_\alpha^{(e)} \asymp N_{\alpha,c-c}^{(e)} \asymp N_{\alpha,p-p}^{(e)} \asymp N_{\alpha,c-p}^{(e)} \asymp \alpha^2 \text{ almost surely as } \alpha \to \infty.
$$

If we define $Q_{\alpha,c} = \sum_i \mathbb{1}_{w_{i1}>0}\mathbb{1}_{\theta_i \le \alpha}$ then $Q_{\alpha,c}$ is a homogeneous Poisson process on $\mathbb{R}_+$ with intensity $\int_{(0,\infty)^K} \rho(dw_1, dw_2, \dots, dw_K) < \infty$ (by assumption (2.9)). By the the law of large numbers, $Q_{\alpha,c} \sim \alpha \int_{(0,\infty)^K} \rho(dw_1, dw_2, \dots, dw_K)$. As

$$
\sqrt{N_{\alpha,c-c}^{(e)}} \le N_{\alpha,c} \le Q_{\alpha,c}
$$

it follows that, almost surely as $\alpha \to \infty$,

$$
N_{\alpha,c} \asymp \alpha.
$$

Similarly, if $\int_{\{0\}\times(0,\infty)^K} \rho(dw_1, dw_2, \dots, dw_K) < \infty$, then $N_{\alpha,p} \asymp \alpha$ and therefore $N_\alpha \asymp \alpha$. Otherwise, define

$$
\widetilde{N}_{\alpha,p} = \sum_i \mathbb{1}_{w_{i1}=0}\mathbb{1}_{\theta_i \le \alpha}\mathbb{1}_{\sum_j \mathbb{1}_{w_{i1}=0} z_{ij}\mathbb{1}_{\theta_j \le \alpha}>0}
$$

as the number of periphery nodes with at least one connection to a periphery node. Note that

$$
N_{\alpha,p} \ge \widetilde{N}_{\alpha,p}.
$$

Two nodes i and j in the periphery connect with probability

$$\begin{cases} 1-e^{-2\sum_{k=2}^{K} w_{ik}w_{jk}} & i \neq j \\ 1-e^{-\sum_{k=2}^{K} w_{ik}^2} & i=j \end{cases} \tag{A.1}$$

where $(w_{i2},\dots,w_{iK})_{i\geq 1, w_{i1}=0}$ are the points of a Poisson point process with mean $\rho_p(dw_2,\dots,dw_K) = \int_{w_1\in[0,\infty)} \rho_p(dw_1,dw_2,\dots,dw_K)$. This is therefore the same model as that of [41]. We therefore have, using Proposition 4 in [41]

$$\widetilde{N}_{\alpha,p} = \omega(\alpha) \text{ if } \int \rho_p(dw_2,\dots,dw_K) = \infty \tag{A.2}$$

and therefore $N_{\alpha,p} = \omega(\alpha)$ and $N_\alpha = \omega(\alpha)$. Noting that $\int \rho_p(dw_2,\dots,dw_K) = \int_{\{0\}\times(0,\infty)^{K-1}} \rho(dw_1,dw_2,\dots,dw_K)$ finishes the proof of Theorem (3.1).

We now consider the particular case of the Lévy measure (2.11). We have

$$\rho_p(dw_2,\dots,dw_K) = \int_0^\infty e^{-w_0} w_0^{-(K-1)} F\left(\frac{dw_2}{w_0},\dots,\frac{dw_K}{w_0}\right)\rho_0(dw_0). \tag{A.3}$$

where F is a product of independent gamma distributions, and ρ_0 is the jump part of a GGP with parameters σ and τ. We recognise this as the particular compound CRM model of [41], except that we now have a base measure

$$\rho_{0,p}(dw_0) = e^{-w_0}\rho_0(dw_0)$$

Then $\int \rho_0(dw_0) = \infty \implies \int \rho_{0,p}(dw_0) = \infty$ for our particular choice of ρ_0. Hence, the results of Proposition 5 of [41] tell us that

$$\widetilde{N}_{\alpha,p} = \Omega(\alpha^{1+\sigma})$$

Since $N_\alpha \geq N_{\alpha,p} \geq \widetilde{N}_{\alpha,p}$ we also have that

$$N_{\alpha,p} = \Omega(\alpha^{1+\sigma}), \qquad N_\alpha = \Omega(\alpha^{1+\sigma})$$

Returning to the core nodes, we know that $N_{\alpha,c} \asymp \alpha$ if

$$\int_{(0,\infty)^K} \rho(dw_1,dw_2,\dots,dw_K) < \infty.$$

But $\int_{(0,\infty)^K} \rho(dw_1,dw_2,\dots,dw_K) = \int \rho_c(dw_2,\dots,dw_K)$, where

$$\rho_c(dw_2,\dots,dw_K) = \int_{w_1\in[0,\infty)} \rho_c(dw_1,dw_2,\dots,dw_K).$$

For the particular case of the Lévy measure (2.11), we have

$$\rho_c(dw_2,\dots,dw_K) = \int_0^\infty \left(1-e^{-w_0}\right) w_0^{-(K-1)} F\left(\frac{dw_2}{w_0},\dots,\frac{dw_K}{w_0}\right)\rho_0(dw_0). \tag{A.4}$$

As before, this is the same as the particular compound CRM model of [41], except that we now have a base measure

$$\rho_{0,c}(dw_0) = \left(1 - e^{-w_0}\right)\rho_0(dw_0)$$

In this case $\int \rho_{0,c}(dw_0)$ is the Laplace exponent $\psi(1)$ of the base Lévy measure ρ_0, which is finite. Hence, the results of Proposition 5 of [41] tell us that $\int_{(0,\infty)^K} \rho(dw_1, dw_2, \ldots, dw_K) < \infty$ as desired, and thus $N_{\alpha,c} \asymp \alpha$. This completes the proof of Theorem 3.2.

In the particular case $K = 2$, we can give a more direct proof of the asymptotics for the periphery nodes and overall graph. We present this here for ease of understanding. In this case we have

$$\begin{aligned}\rho_c(dw_1, dw_2) &= \rho(dw_1, dw_2)\mathbb{1}_{w_1>0}\\ \rho_p(dw_1, dw_2) &= \rho(dw_1, dw_2)\mathbb{1}_{w_1=0}.\end{aligned}$$

Using a slight abuse of notation, let $\rho_p(w_2)$ and $\rho_0(w_0)$ denote the intensity functions of the measures $\rho_p(dw_2)$ and $\rho_0(dw_0)$. Let f denote the pdf of a gamma random variable with parameters a and b. We have

$$\rho_p(w_2) = \int_0^\infty e^{-w_0} w_0^{-1} f(w_2/w_0)\rho_0(w_0)dw_0. \tag{A.5}$$

As $e^{-w_0}\rho_0(w_0) \sim w_0^{-1-\sigma}\Gamma(1-\sigma)^{-1}$ as w_0 tends to 0 when $\sigma \in (0,1)$ we have, using [1, Corollary 6] (see also [3, Theorem 4.1.6 page 201]),

$$\rho_p(w_2) \sim w_2^{-1-\sigma}\Gamma(1-\sigma)^{-1}\mathbb{E}[\beta_2^\sigma]$$

where β_2 has cdf F. Finally, using [3, Proposition 1.5.8. p. 26], we obtain

$$\int_x^\infty \rho_p(w_2)dw_2 \sim x^{-\sigma}\sigma^{-1}\Gamma(1-\sigma)^{-1}\mathbb{E}[\beta_2^\sigma]$$

and it follows from [6, Theorem 4] that

$$\widetilde{N}_{\alpha,p} = \Omega(\alpha^{1+\sigma})$$

Since $N_\alpha \geq N_{\alpha,p} \geq \widetilde{N}_{\alpha,p}$ we also have that

$$N_{\alpha,p} = \Omega(\alpha^{1+\sigma}), \qquad N_\alpha = \Omega(\alpha^{1+\sigma})$$

which gives the desired results for the periphery nodes and the overall graph. The corresponding results for the core nodes, which tell us that $N_{\alpha,c} \asymp \alpha$, can be found by taking $K = 2$ in the general proof. This then completes the proof of Theorem 3.2 in the particular case $K = 2$.

The results of Corollary 3.2.1, both follow directly from $N_{\alpha,c} \asymp \alpha$ and the fact that $N_\alpha = \Omega\left(\alpha^{1+\sigma}\right)$ for the overall graph.

Appendix B: More simulation results

We show here empirical results on the effects that changing various parameters of the model have on the degree distributions, sparsity properties and core proportion. We restrict ourselves to the $K = 2$ case here, for ease of visualisation.

B.1. Degree distributions

We first look at the degree distributions for the overall graph, and for the core and periphery nodes separately. The first thing we notice is that, as we expect, the core and periphery nodes have very different degree distributions. In Figure 18 we see the results for varying σ and τ. Here we see that increasing σ leads to lower degree nodes in the overall graph, as well as both core and periphery regions. We expect this, since we know that a larger value of σ means that the graph is more sparse. The distribution begins to look closer to a pure power-law for large σ. We see further that increasing τ has little effect on the degree distribution in the core (for $\tau = 5$ the core size is very small, leading to less interpretable results). We also see fewer nodes with a high degree, and again behaviour more closely resembling a power-law.

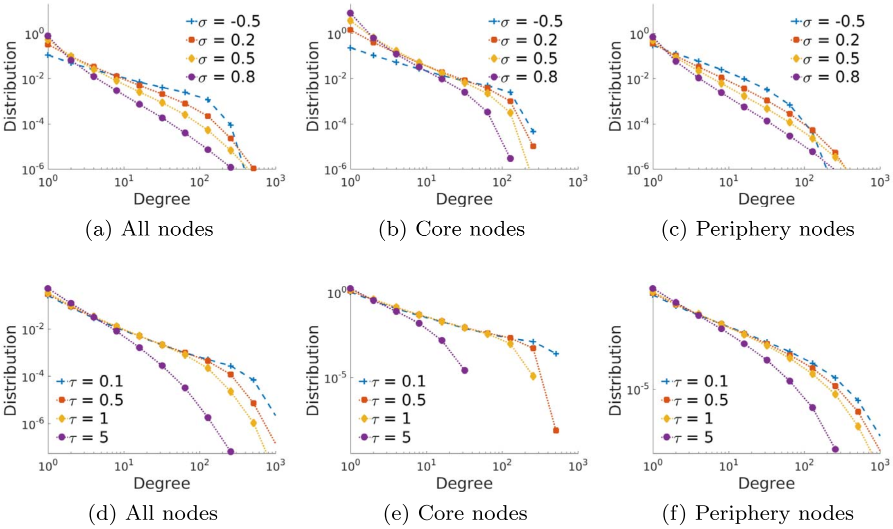
Figure 19 shows the results for varying a and b. When increasing a we see that the shape of the degree distribution for the core does not change, although the number of nodes in the core increases. Overall and in the periphery we see little difference in the degree distribution apart from nodes with high degree. For larger values of a there are more of these. The degree distributions for b are similar, except that increasing b decreases the number of nodes with a high degree.

B.2. Sparsity

We have seen how the sparsity properties in the different regions are controlled by σ. From Figures 20 and 21 we see that, as expected, changing the parameters a and b does not affect this, since the gradients of the lines in each case are the same. However, we do see that for fixed sized graphs, increasing a increases the density in the overall graph, and in the different regions. Conversely, increasing b decreases the density in the core-periphery and periphery regions. The density in the core region is also decreased, but this effect appears to be far less significant.

B.3. Core proportion

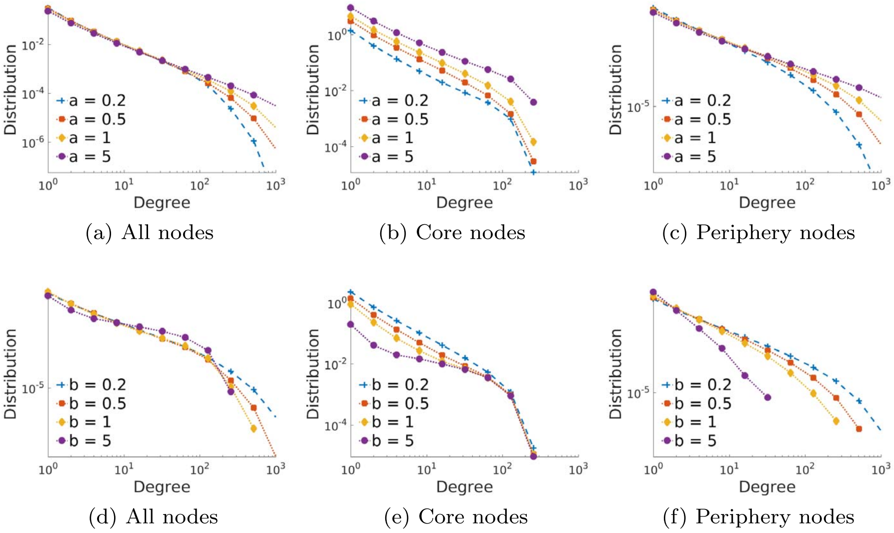
We are finally interested in how the proportion of nodes in the core is affected by changing a and b. From Figure 22 we see that, as expected, the asymptotic rates are not affected by changing these hyperparameters. However, for fixed size graphs, increasing a decreases the relative size of the core region, whilst the opposite is true for b.



(a) All nodes (b) Core nodes (c) Periphery nodes

(d) All nodes (e) Core nodes (f) Periphery nodes

Fig 18. *Degree distributions of (left) all, (middle) core and (right) periphery nodes for graphs generated from the SparseCP model with varying values of (top row) σ and (bottom row) τ.*



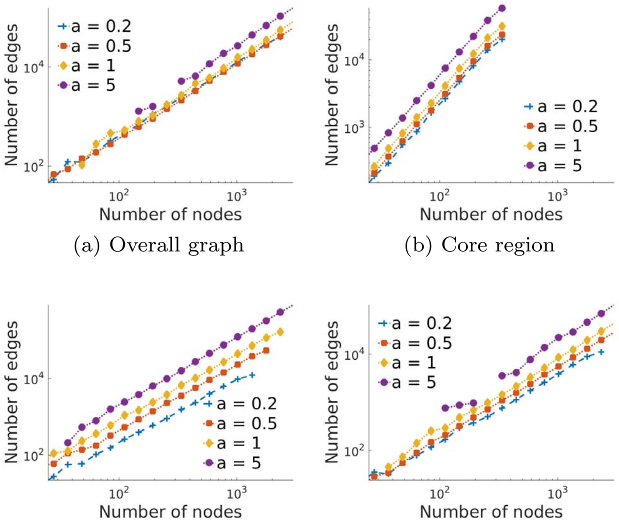
(a) All nodes (b) Core nodes (c) Periphery nodes

(d) All nodes (e) Core nodes (f) Periphery nodes

Fig 19. *Degree distributions of (left) all, (middle) core and (right) periphery nodes for graphs generated from the SparseCP model with varying values of (top row) a and (bottom row) b.*

Appendix C: MCMC algorithm details

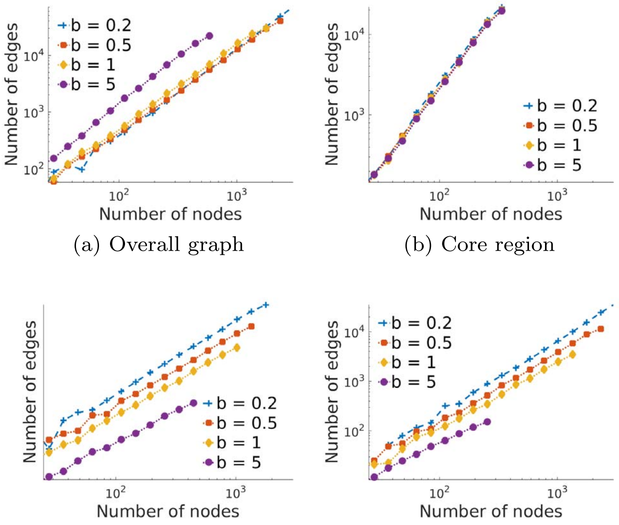
We give here the details of our MCMC algorithm, as detailed in Algorithm 1, in particular how it differs from the Algorithm of [41].



(a) Overall graph (b) Core region

(c) Core-periphery region (d) Periphery region

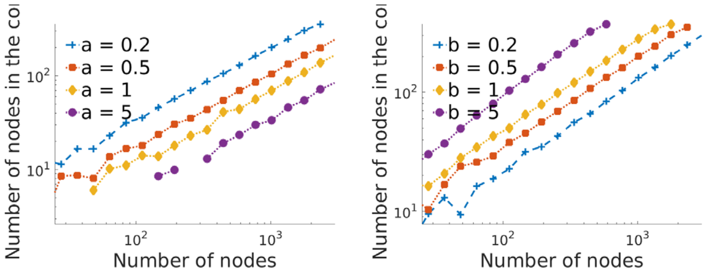
Fig 20. *Relationship between nodes and edges for the (a) overall graph, (b) core, (c) core-periphery region and (d) periphery of the graph generated from the SparseCP model with varying a.*



(a) Overall graph (b) Core region

(c) Core-periphery region (d) Periphery region

Fig 21. *Relationship between nodes and edges for the (a) overall graph, (b) core, (c) core-periphery region and (d) periphery of the graph generated from the SparseCP model with varying b.*



(a) Varying a (b) Varying b

FIG 22. *Relative core size for graphs generated from the SparseCP model with (a) varying a and (b) varying b.*

As in [41], we assume that we have observed a set of connections $(z_{ij})_{1\leq i,j,\leq N_\alpha}$ where N_α is the number of nodes with at least one connection. We want to infer the parameters $(w_{i1},\dots,w_{iK})_{i=1,\dots,N_\alpha}$. We also want to estimate the sums of the parameters for the nodes with no connection $(w_{*1},\dots,w_{*K})$, the hyperparameters ϕ of the mean intensity ρ and the parameter α which is also assumed to be unknown. Thus the aim is to sample from the posterior

$$p((w_{1k},\dots,w_{N_\alpha k},w_{*k})_{k=1,\dots,K},\phi,\alpha \mid (z_{ij})_{1\leq i,j,\leq N_\alpha}) \tag{C.1}$$

As in [41], we introduce the latent count variables $\tilde{n}_{ijk}$ using the data augmentation scheme given in Section C.3. Then we can define an algorithm which uses Metropolis-Hastings (MH) and Hamiltonian Monte Carlo (HMC) updates with a Gibbs sampler to perform posterior inference. At each iteration of the Gibbs sampler we update using Algorithm 1.

Algorithm 1 MCMC sampler for posterior inference.

```
At each iteration:
1: Update $(w_{i0},\beta_{i1},\dots,\beta_{iK})_{i=1,\dots,N_\alpha}$ given the rest at the same time using HMC.
2: Update the hyperparameters $(\phi,\alpha)$ and the total masses $(w_{*1},\dots,w_{*K})$ given the rest using MH.
3: Update the latent variables $\tilde{n}_{ijk}$ given the rest using (C.9).
```

C.1. Step 1

In the first step of Algorithm 1, we use the conditional distribution given the latent variable counts as defined in [41]:

$$p((w_{10},\dots,w_{N_\alpha 0}),(\beta_{1k},\dots,\beta_{N_\alpha k},w_{*k})_{k=1,\dots,K} \mid (n_{ijk})_{1\leq i,j,\leq N_\alpha,k=1,\dots,K},\phi,\alpha)$$
$$\propto \left[\prod_{i=1}^{N_\alpha} w_{i0}^{m_i}\right]\left[\prod_{i=1}^{N_\alpha}\prod_{k=1}^{K}\beta_{ik}^{m_{ik}}\right] e^{-\sum_{k=1}^{K}(w_{*k}+\sum_{i=1}^{N_\alpha} w_{i0}\beta_{ik})^2}$$

$$\times \left[\prod_{i=1}^{N_\alpha} f(\beta_{i2},\dots,\beta_{iK};\phi)\right]\left[\prod_{i=1}^{N_\alpha}(1-e^{-w_{i0}})^{\beta_{i1}}e^{-w_{i0}(1-\beta_{i1})}\right]$$
$$\times \left[\prod_{i=1}^{N_\alpha}\rho_0(w_{i0};\phi)\right]\alpha^{N_\alpha}g_{*\alpha}(w_{*1},\dots,w_{*K};\phi) \tag{C.2}$$

where $m_i = \sum_{k=1}^K m_{ik}$, $m_{ik} = \sum_{j=1}^{N_\alpha} \tilde{n}_{ijk}$ and f is the density function corresponding to the distribution F (in this case the product of gamma densities). The key difference in our case compared to [41], as we see in Section 2.2, is that the β_{i1} are not identically distributed, and depend on w_{i0}. This gives us the separate term involving the β_{i1} in (C.2). However, we see that we can still follow the same algorithm, with some modifications.

In general, if ρ can be evaluated pointwise, a MH update can be used for this step, however this will scale poorly with the number of nodes. In the compound CRM case, if f and ρ_0 are differentiable then a HMC update can be used. For the SNetOC model where f is the product of independent gamma distributions and ρ_0 is the mean measure of a generalized gamma process this differentiability condition does hold. Thus, a HMC sampler is used to update $(w_{i0},\beta_{i1},\dots,\beta_{iK})_{i=1,\dots,N_\alpha}$ all at once, with the potential energy function U defined as

$$U(w_0,\beta) = \log\left(p((\log(w_{i0}))_{i=1,\dots,N_\alpha},(\log(\beta_{ik}))_{i=1,\dots,N_\alpha;k=1,\dots,K}\mid rest)\right) \tag{C.3}$$

$U(w_0,\beta)$ can be calculated from (C.2) using a simple change of variables. Furthermore its derivatives have simple analytic forms, and so the HMC step can be computed exactly.

In our case this strategy will not work, because β_1 is a binary variable, and so both the log transformation and the HMC update of the $(\beta_{i1})_{i=1,\dots,N_\alpha}$ cannot be used. We propose an alternative solution, using a Gibbs step to update β_1 separately from the others as follows

1. Update $(w_{i0},\beta_{i2},\dots,\beta_{iK})_{i=1,\dots,N_\alpha}$ from $p((w_{i0},\beta_{i2},\dots,\beta_{iK})_{i=1,\dots,N_\alpha}\mid(\beta_{i1})_{i=1,\dots,N_\alpha},\mathrm{rest})$
2. Update $(\beta_{i1})_{i=1,\dots,N_\alpha}$ from $p((\beta_{i1})_{i=1,\dots,N_\alpha}\mid(w_{i0},\beta_{i2},\dots,\beta_{iK})_{i=1,\dots,N_\alpha},\mathrm{rest})$

This method also allows us to deal with the fact that the β_{i1} are not identically distributed, and depend on the w_{i0}. The two conditional distributions can be found from (C.2) as follows. Ignoring the terms not involving $(w_0,\beta_2,\dots,\beta_K)$, the first conditional is proportional to

$$\left[\prod_{i=1}^{N_\alpha} w_{i0}^{m_i}\right]\left[\prod_{i=1}^{N_\alpha}\prod_{k=2}^{K}\beta_{ik}^{m_{ik}}\right]e^{-\sum_{k=1}^K(w_{*k}+\sum_{i=1}^{N_\alpha}w_{i0}\beta_{ik})^2}\left[\prod_{i=1}^{N_\alpha}\rho_0(w_{i0};\phi)\right]$$
$$\times\left[\prod_{i=1}^{N_\alpha}\prod_{k=2}^{K}\beta_{ik}^{a_k-1}e^{-b_k\beta_{ik}}\frac{b_k^{a_k}}{\Gamma(a_k)}\right]\left[\prod_{i=1}^{N_\alpha}(1-e^{-w_{i0}})^{\beta_{i1}}e^{-w_{i0}(1-\beta_{i1})}\right] \tag{C.4}$$

We can simulate from this using HMC as before, except that now β_1 is considered to be constant, and we have to incorporate an extra term involving the w_{i0} coming from the distribution of $\beta_{i0} \mid w_{i0}$. The details of the exact HMC sampler are therefore very similar to the SNetOC model, and we omit them here.

The second conditional distribution is proportional to

$$\left[\prod_{i=1}^{N_\alpha} \beta_{i1}^{m_{i1}}\right] e^{-(w_{*1}+\sum_{i=1}^{N_\alpha} w_{i0}\beta_{i1})^2} \left[\prod_{i=1}^{N_\alpha} (1-e^{-w_{i0}})^{\beta_{i1}} e^{-w_{i0}(1-\beta_{i1})}\right] \tag{C.5}$$

In order to simulate from this distribution we first approximate the conditional by linearizing the exponent as follows

$$\begin{aligned} &\exp\left\{-\left(w_{*1}+\sum_{i=1}^{N_\alpha} w_{i0}\beta_{i1}\right)^2\right\} \\ &\approx \exp\left\{-\left(w_{*1}+\sum_{i=1}^{N_\alpha} w_{i0}\beta_{i1}^*\right)\left(w_{*}+\sum_{i=1}^{N_\alpha} w_{i0}\beta_{i1}\right)\right\} \end{aligned}$$

where β_{i1}^* is the sampled value of β_{i1} from Step 1 of the previous iteration of the overall Gibbs sampler in Algorithm 1. We thus treat β_{i1}^* as a constant here. Using this approximation, and defining

$$c_1 = \left(w_{*1}+\sum_{i=1}^{N_\alpha} w_{i0}\beta_{i1}^*\right)$$

we have that the conditional distribution of $(\beta_{i1})_{i=1,\dots,N_\alpha}$ given the rest is proportional to

$$\begin{aligned} &\exp\left\{-c_1\sum_{i=1}^{N_\alpha} w_{i0}\beta_{i1}\right\}\left[\prod_{i=1}^{N_\alpha} \beta_{i1}^{m_{i1}}(1-e^{-w_{i0}})^{\beta_{i1}} e^{-w_{i0}(1-\beta_{i1})}\right] \\ &= \prod_{i=1}^{N_\alpha} \beta_{i1}^{m_{i1}}\left[e^{-c_1 w_{i0}}(1-e^{-w_{i0}})\right]^{\beta_{i1}} e^{-w_{i0}(1-\beta_{i1})} \\ &= \prod_{i=1}^{N_\alpha} f_i(\beta_{i1}) \end{aligned}$$

We see that we can simulate each of the β_{i1} individually. The distribution $f_i(\beta_{i1})$ depends on m_{i1}, and there are two cases.

1. If $m_{i1} > 0$ then, recalling that β_{i1} is a binary variable, $Pr(\beta_{i1}=0) = f_i(0) = 0$ and so $Pr(\beta_{i1}=1)=1$. This means that if the sum of the latent counts $\sum_j \tilde{n}_{ij1}$ from the previous iteration is not equal to 0 then β_{i1} will be updated to 1. We also know from (C.9) that if $\beta_{i1}=0$ then $\tilde{n}_{ij1}=0 \quad \forall j$. However, this does not lead to a problem of losing irreducibility of the Markov chain, since the reverse is not true.

2. If $m_{i1} = 0$ then we have that

$$f_i(\beta_{i1}) \propto \left[e^{-c_1 w_{i0}}(1 - e^{-w_{i0}})\right]^{\beta_{i1}} e^{-w_{i0}(1-\beta_{i1})}$$

which we recognize as a Bernoulli distribution with parameter

$$p_1 = \frac{1 - e^{-w_{i0}}}{1 - e^{-w_{i0}} + e^{-(1-c_1)w_{i0}}}$$

Thus we can update the β_{i1} given the rest by sampling

$$\beta_{i1} \sim \mathrm{Ber}(p_1)$$

Of course, we are conditioning on m_{i1} when performing this step, so we will always know what case we are in. Thus, the above can be used to sample β_1 from the relevant conditional distribution.

C.2. Step 2

In this step we want to sample from $p((w_{*k})_{k=1,\ldots,K}, \phi, \alpha \mid rest)$, which we know from (C.2) is proportional to

$$p(\phi)p(\alpha)e^{-\sum_{k=1}^{K}(w_{*k}+\sum_{i=1}^{N_\alpha} w_{ik})^2}\left[\prod_{i=1}^{N_\alpha}\rho(w_{i1},\ldots,w_{iK};\phi)\right] \\ \times \alpha^{N_\alpha} g_{*\alpha}(w_{*1},\ldots,w_{*K};\phi)$$

This distribution is not of a standard form, and involves the pdf $g_{*\alpha}(w_{*1},\ldots,w_{*K};\phi)$ of the total masses. This pdf typically has no analytic expression, but its Laplace transform is available. For the SNetOC model a MH step is therefore used, with a proposal

$$\begin{aligned} q(\tilde{w}_{*1:K}, \tilde{\phi}, \tilde{\alpha} \mid w_{*1:K}, \phi, \alpha) =& q(\tilde{w}_{*1:K} \mid w_{*1:K}, \tilde{\phi}, \tilde{\alpha}) \times q(\tilde{\phi} \mid \phi) \\ & \times q(\tilde{\alpha} \mid w_{*1:K}, \tilde{\phi}, \alpha) \end{aligned} \tag{C.6}$$

where

$$q(\tilde{\alpha} \mid w_{*1:K}, \tilde{\phi}, \alpha) = \Gamma(\tilde{\alpha}; a_\alpha + N_\alpha, b_\alpha + \psi(\lambda_1,\ldots,\lambda_K;\tilde{\phi})) \tag{C.7}$$

and $q(\tilde{w}_{*1:K} \mid w_{*1:K}, \tilde{\phi}, \tilde{\alpha})$ is an exponentially tilted version of $g_{*\alpha}$. Here, $\lambda_k = w_{*k} + 2\sum_{i=1}^{N_\alpha} w_{ik}$, and ψ is the Laplace exponent. In the compound CRM case ψ takes a simple form that only involved evaluating a one-dimensional integral. The calculation of ψ in our case can be done similarly, the only difference being that we need the moment generating function of a Bernoulli distribution for the first component. The details are omitted here. Similarly, since the distribution of β_1 has no hyperparameter, the same $q(\tilde{\phi} \mid \phi)$ can be used in our case as the compound CRM case of the SNetOC model, ignoring the $k = 1$ term.

The other challenging part of the step is to simulate $w_{*1:K}$ from the distribution $q(\tilde{w}_{*1:K} \mid w_{*1:K}, \tilde{\phi}, \tilde{\alpha})$. Again we can do this as in the compound CRM case of [41], by setting $(w_{*1}, \dots, w_{*K}) = X_\epsilon + X^\epsilon$, with $X_\epsilon = \sum_{i|w_{i0}<\epsilon} w_{i0}(\beta_{i1}, \dots, \beta_{iK})$ and $X^\epsilon = \sum_{i|w_{i0}>\epsilon} w_{i0}(\beta_{i1}, \dots, \beta_{iK})$. Then, a realization of X^ϵ can be simulated exactly from a Poisson process with mean measure

$$\alpha e^{-w_0 \sum_{k=1}^K (\gamma_k + \lambda_k)\beta_k} f(\beta_1, \dots, \beta_K) \rho_0(w_0) \mathbb{1}_{w_0 > \epsilon} \tag{C.8}$$

This is done using an adaptive thinning procedure as detailed in Appendix D of [41]. We can use the same approach here, using the same adaptive bound by noting that $(1 - e^{-w_0}) \leq 1$. X_ϵ can be approximated as before using a truncated Gaussian random vector. All that changes are the exact forms of the mean and variance of the approximating random vector, and we omit the details here.

C.3. Step 3

This step can be done as in [41] by introducing the latent variables $\widetilde{n}_{ijk} = n_{ijk} + n_{jik}$ where

$$(\widetilde{n}_{ij1}, \dots, \widetilde{n}_{ijK}) \,|z, w \sim \begin{cases} \delta_{(0)} & \text{if } z_{ij} = 0 \\ \text{tPoisson}\left(2w_{i1}w_{j1}, \dots, 2w_{iK}w_{jK}\right) & \text{if } z_{ij} = 1,\ i < j \end{cases} \tag{C.9}$$

$$\left(\frac{\widetilde{n}_{ii1}}{2}, \dots, \frac{\widetilde{n}_{iiK}}{2}\right) |z, w \sim \text{tPoisson}\left(w_{i1}^2, \dots, w_{iK}^2\right) \text{ if } z_{ii} = 1$$

By convention we set $\widetilde{n}_{ijk} = \widetilde{n}_{jik}$ for all $i > j$, and we have that $m_{ik} = \sum_j \widetilde{n}_{ijk}$. The pmf of a tPoisson$(\lambda_1, \dots, \lambda_K)$ distribution is

$$\text{tPoisson}(x_1, \dots, x_K; \lambda_1, \dots, \lambda_K) = \frac{\prod_{k=1}^K \text{Poisson}(x_k; \lambda_k)}{1 - \exp(-\sum_{k=1}^K x_k \lambda_k)} \mathbb{1}_{\{\sum_{k=1}^K x_k > 0\}}$$

which can be sampled from by sampling $x = \sum_{k=1}^K x_k$ from a univariate zero-truncated Poisson distribution [37] with rate $\sum_{k=1}^K \lambda_k$, and then sampling

$$(x_1, \dots, x_K) \mid (\lambda_1, \dots, \lambda_K), x \sim \text{Multinomial}\left(x, \left(\frac{\lambda_1}{\sum \lambda_k}, \dots, \frac{\lambda_K}{\sum \lambda_k}\right)\right).$$

The only difference in our case is that λ_1 may be 0, in which case we set $x_1 = 0$.

In the update for β_1, we see that if $m_{i1} > 0$ then β_{i1} is set identically to 1, because there is no posterior mass at 0. In order to get better mixing, we can instead update the $\widetilde{n}_{ijk}$ via a Metropolis-Hastings step that proposes $m_{i1} = 0$ more often. We do this as follows

1. Choose a set $I = \{i_1, \dots, i_N\}$, $i_1 < \dots < i_N$, with $N = |I|$ of indices for which we will propose $m_{i1} = 0\ \forall i \in I$.

2. Calculate the set A of edges (i,j) such that $\widetilde{n}_{ij1} = 0 \; \forall (i,j) \in A \iff m_{i1} = 0 \; \forall i \in I$.
3. For $(i,j) \notin A$, update $(\widetilde{n}_{ij1}, \dots, \widetilde{n}_{ijK})$ as normal, using the conditional distribution given z, w.
4. For $(i,j) \in A$, propose an update for $(\widetilde{n}_{ij1}, \dots, \widetilde{n}_{ijK})$ from the mixture distribution:

$$\begin{aligned} p_{lat} \left(\mathbb{1}(\widetilde{n}_{ij1} = 0) \times \text{tPoisson} \left(2w_{i2}w_{j2}, \dots, 2w_{iK}w_{jK} \right) \right) \\ +(1 - p_{lat}) \left(\text{tPoisson} \left(2w_{i1}w_{j1}, \dots, 2w_{iK}w_{jK} \right) \right) \end{aligned}$$

   for $i \neq j$, and similarly for $i = j$. This means that with probability p_{lat} we set $\widetilde{n}_{ij1}$ to 0 and simulate $\widetilde{n}_{ij2}, \dots, \widetilde{n}_{ijK}$ from a truncated Poisson distribution. With probability $1 - p_{lat}$, $\widetilde{n}_{ij1}, \dots, \widetilde{n}_{ijK}$ are simulated from a truncated Poisson distribution as normal.
5. Accept the proposal using the standard Metropolis Hastings acceptance rate given by

$$\alpha = \min \left(1, \frac{P(\widetilde{n})}{P(\widetilde{n}_{old})} \frac{Q(\widetilde{n}_{old})}{Q(\widetilde{n})} \right)$$

   where $\widetilde{n} = (\widetilde{n})_{1 \le i,j \le N_\alpha, k=1,\dots,K}$, P is the true distribution of the $\widetilde{n}_{ijk}$ as given in (C.9), Q is the proposal distribution detailed above and $\widetilde{n}_{old}$ is the value of $\widetilde{n}$ from the previous iteration of the overall sampler.

In practice, we find that this method allows for faster mixing of the algorithm.

C.4. Parameter initialization

We start with the initialization method employed by [41], which performs a short run with $K = 1$, using the parameter estimates obtained there as initial values for the full run. When testing our model on simulated data sets with $K = 2$ (i.e. with a core parameter w_1 and overall sociability parameter w_2) we find that when the core-periphery structure is particularly weak our estimated values of w_1 are close to the true values of w_2, and vice versa. Similarly, when taking $K > 2$, we sometimes find that the estimated values of the core parameter are instead close to the values of one of the community parameters.

We therefore introduce a new initialization procedure, in order to ensure that our core parameter w_1 is estimating the core-periphery structure. We first perform a short initialization run using the SNetOC model. Having done this, we use the community sociabilities from the SNetOC model in order to initialize the weights $w_k, \; k = 2, \dots, K$ in our model. Initializing in this way prevents the problem of w_1 approximating the sociabilities of one of the communities. As shown in Section 4, the SNetOC model is not an ideal model for identifying core-periphery structure for several reasons. However, we do see that one of the identified community sociability parameters can sometimes approximate our core parameter, and thus provide us with a way to initialize this parameter as well. In practice, we find that initializing the size of the core to be significantly larger than we believe it to be leads to better mixing of the MCMC sampler.

C.5. Approximation of the log-posterior density

The posterior probability density function, up to a normalizing constant, takes the form

$$\begin{aligned} & p\left((w_{i0},\beta_{i1},\dots,\beta_{iK})_{i=1,\dots,N_\alpha},\phi,\alpha \mid (z_{ij})_{1\leq i,j\leq N_\alpha}\right) \\ & \propto \left[\prod_{i=1}^{N_\alpha}\prod_{j=1}^{N_\alpha}\left(\frac{1-e^{-\sum_k w_{i0}\beta_{ik}w_{j0}\beta_{jk}}}{e^{-\sum_k w_{i0}\beta_{ik}w_{j0}\beta_{jk}}}\right)^{z_{ij}}\right] e^{-\sum_{k=1}^{K}(w_{*k}+\sum_{i=1}^{N_\alpha} w_{i0}\beta_{ik})^2} \\ & \quad \times \left[\prod_{i=1}^{N_\alpha}\prod_{k=2}^{K}\beta_{ik}^{a_k-1}e^{-b_k\beta_{ik}}\frac{b_k^{a_k}}{\Gamma(a_k)}\right]\left[\prod_{i=1}^{N_\alpha}(1-e^{-w_{i0}})^{\beta_{i1}}e^{-w_{i0}(1-\beta_{i1})}\right] \\ & \quad \times \left[\prod_{i=1}^{N_\alpha}\rho_0(w_{i0};\phi)\right]\alpha^{N_\alpha}p(\phi,\alpha)g_{*\alpha}(w_{*1},\dots,w_{*K};\phi) \end{aligned} \tag{C.10}$$

where $m_{ik}=\sum_{j=1}^{N_\alpha} n_{ijk}+n_{jik}$ and $g_{*\alpha}(w_{*1},\dots,w_{*K};\phi)$ is the probability density function of the random vector $(W_1\left([0,\alpha]\right),\dots,W_K\left([0,\alpha]\right))$.

This is intractable due to the lack of an analytic expression for $g_{*\alpha}$. We can however approximate the log-posterior. Noting that $w_{*k}=o(\sum_{i=1}^{N_\alpha} w_{ik})$ as $\alpha\to\infty$, we have

$$(w_{*k}+\sum_{i=1}^{N_\alpha} w_{ik})^2 \simeq (\sum_{i=1}^{N_\alpha} w_{ik})^2 + 2w_{*k}\sum_{i=1}^{N_\alpha} w_{ik}.$$

Using this approximation, one can now integrate out w_{*k} and we then obtain the approximation

$$\begin{aligned} & p((w_{1k},\dots,w_{N_\alpha k})_{k=1,\dots,K},\phi,\alpha \mid (z_{ij})_{1\leq i,j\leq N_\alpha}) \\ & \simeq\propto \left[\prod_{i=1}^{N_\alpha}\prod_{j=1}^{N_\alpha}\left(\frac{1-e^{-\sum_k w_{i0}\beta_{ik}w_{j0}\beta_{jk}}}{e^{-\sum_k w_{i0}\beta_{ik}w_{j0}\beta_{jk}}}\right)^{z_{ij}}\right] e^{-\sum_{k=1}^{K}(\sum_{i=1}^{N_\alpha} w_{i0}\beta_{ik})^2} \\ & \quad \times \left[\prod_{i=1}^{N_\alpha}\prod_{k=2}^{K}\beta_{ik}^{a_k-1}e^{-b_k\beta_{ik}}\frac{b_k^{a_k}}{\Gamma(a_k)}\right]\left[\prod_{i=1}^{N_\alpha}(1-e^{-w_{i0}})^{\beta_{i1}}e^{-w_{i0}(1-\beta_{i1})}\right] \\ & \quad \times \left[\prod_{i=1}^{N_\alpha}\rho_0(w_{i0};\phi)\right]\alpha^{N_\alpha}\exp\left[-\alpha\psi\left(2\sum_{i=1}^{N_\alpha} w_{i1},\dots,2\sum_{i=1}^{N_\alpha} w_{iK};\phi\right)\right]p(\phi,\alpha) \end{aligned} \tag{C.11}$$

where $\psi(t_1,\dots,t_K)$ is the multivariate Laplace exponent, which can be evaluated numerically. We will use the approximated log-posterior density in order to evaluate the convergence of our MCMC sampler in the experiments we run. In particular, we can use it as a one-dimensional statistic from which to calculate the Gelman-Rubin convergence diagnostic [16].

Appendix D: Additional experimental results

In this section, we provide a number of additional plots related to the experiments in Section 4. They are organised as follows. In Section D.1.1 we present the results from the MCMC sampler used to perform posterior inference for the experiments in Section 4.2. The relevant plots comprise Figures 23-25. In Section D.1.2 we detail an additional experiment performing posterior inference on a simulated graph exhibiting both core-periphery and community structure. The relevant plots comprise Figures 26-30. In Section D.1.3 we give the results for additional experiments, in which posterior inference is performed on graphs which do not exhibit core-periphery structure. The relevant plots comprise Figure 31.

In Section D.2 we present the results from the MCMC sampler used to perform posterior inference for the experiments in Section 4.3. In Section D.2.1 we present the results from the US Airport experiment. The relevant plots comprise Figures 32-34. In Section D.2.2 we present the results from the World Trade experiment. The relevant plots comprise Figures 35-36.

D.1. Simulated data

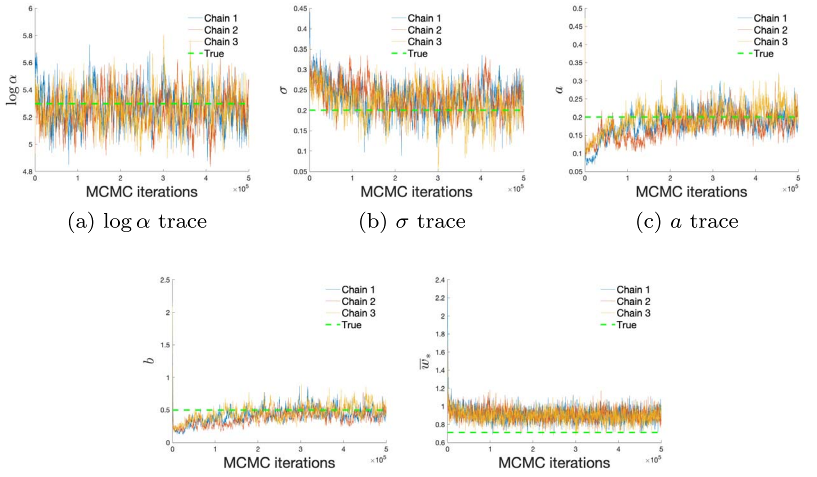
D.1.1. Core-periphery structure only

In Section 4.2, we present a selection of results from performing posterior inference on a simulated graph generated with $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = \frac{1}{K}, a = 0.2$. Here, we present some additional results specifically regarding the output of the MCMC sampler. In Figures 23 and 24 we see the parameter trace plots and histograms.

The green lines and stars respectively correspond to the values of the model parameters used to generate the graphs, and we see that the posterior converges around the true values.

In Figure 25 we give the trace and autocorrelation plots of an approximation of the log-posterior (up to a normalizing constant). The details of the calculation of the approximate log-posterior are given in Section C.5. Here, the green line gives the value of the approximate log-posterior using the true model parameters. The log-posterior trace plot suggests that the chains have converged and the autocorrelation plot shows that the correlation of the samples decreases quickly with increasing lag.

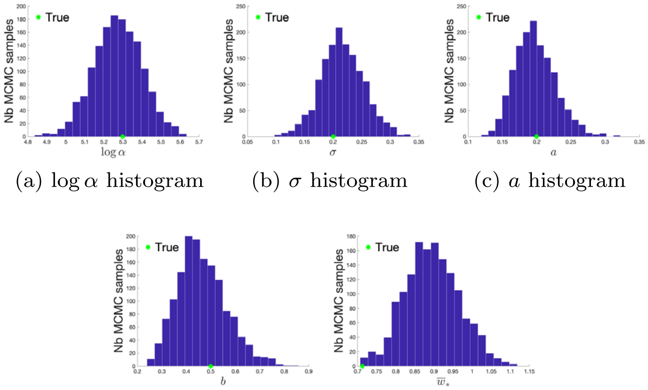
In order to test the convergence of the chains, we calculate the Gelman-Rubin convergence diagnostic $\hat{R}$ [16]. Due to the high number of parameters in the SparseCP model, we calculate a univariate statistic using the sampled values of the approximate log-posterior. Recalling that $\hat{R} < 1.1$ suggests convergence, we find that in our case $\hat{R} = 1.03$. Thus we are satisfied that the chains have indeed converged.



(a) $\log \alpha$ trace (b) σ trace (c) a trace

(d) b trace (e) $\bar{w}_*$ trace

Fig 23. *MCMC trace plots for parameters* (*a*) $\log \alpha$, (*b*) σ, (*c*) a, (*d*) b *and* (*e*) $\bar{w}_*$. *The graph is generated from the SparseCP model with* $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = \frac{1}{K}, a = 0.2$. *The green lines correspond to the values of the parameters used for generating the graph.*



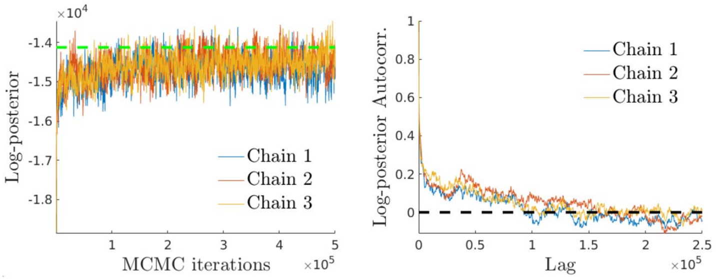
(a) $\log \alpha$ histogram (b) σ histogram (c) a histogram

(d) b histogram (e) $\bar{w}_*$ histogram

Fig 24. *MCMC histograms for parameters* (*a*) $\log \alpha$, (*b*) σ, (*c*) a, (*d*) b *and* (*e*) $\bar{w}_*$. *The graph is generated from the SparseCP model with* $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = \frac{1}{K}, a = 0.2$. *The green stars correspond to the values of the parameters used for generating the graph.*

D.1.2. Core-periphery and community structure

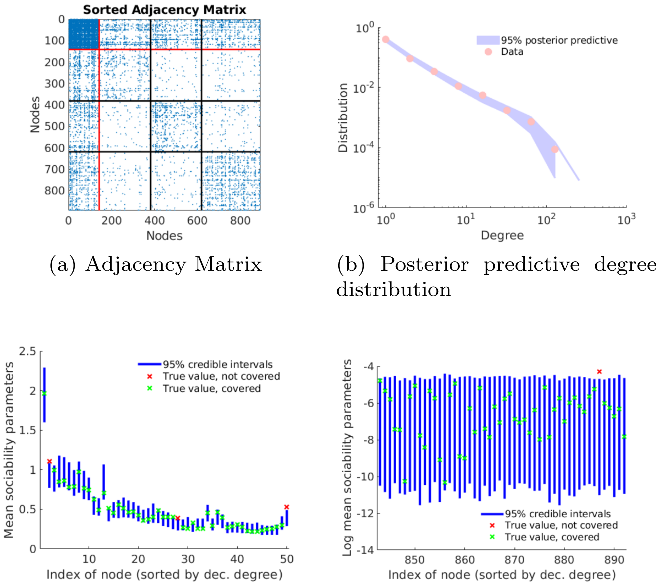
Here, we test our model on a graph with core-periphery and community structure, to confirm that we can detect the presence of both. In this case, K is taken to be 4 in order to generate a graph with a core-periphery structure and 3 overlapping communities. In Figure 26(a) we see the adjacency matrix for the



(a) Log-posterior

(b) Autocorrelation

FIG 25. *Trace plot and autocorrelation of the log-posterior for a graph generated from the SparseCP model with* $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = \frac{1}{K}, a = 0.2$. *The green line in* (a) *corresponds to the value of the approximate log-posterior (up to a constant) under the model parameters used to generate the graph.*



(a) Adjacency Matrix

(b) Posterior predictive degree distribution

(c) Mean sociability credible intervals for high degree nodes

(d) Mean sociability credible intervals for low degree nodes

FIG 26. *Fitting the SparseCP model to simulated data with core-periphery and community structure, generated with* $K = 4, \alpha = 200, \sigma = 0.2, \tau = 1, b_k = b = \frac{4}{K}, a_k = a = 0.2$. (a) *Adjacency matrix, where the nodes are sorted into core and periphery (indicated by the red lines) and then by largest community sociability (indicated by the black lines).* (b) *Posterior predictive degree distribution. When the lower posterior predictive interval bound is 0, we plot only the upper interval bound. Mean sociability* ($\frac{1}{2}(w_{i1} + w_{i2})$) *credible intervals for* (c) *the* 50 *nodes with the highest degrees,* (d) *the* 50 *nodes with the lowest degree. In each case the true values are coloured green if they are covered by the credible interval, and red if not.*

simulated network, sorted into core and periphery (indicated by the red lines) and then by largest community sociability (indicated by the black lines). We see a clear global core-periphery structure and community structure. As before, we see from Figures 26(b) and 26(d) that we are able to recover the degree distribution, as well as the mean sociabilities of the high and low degree nodes.

In Figures 27 and 28 we see the trace plots and histograms for the identifiable parameters for this graph. The green lines and stars correspond to the true values of the parameters, and as before we see that the posterior distribution converges around the true values. In Figure 29 we plot the credible intervals for each of the sociabilities $w_1, \dots, w_4$, and see that we are able to recover both the core and community sociabilities.

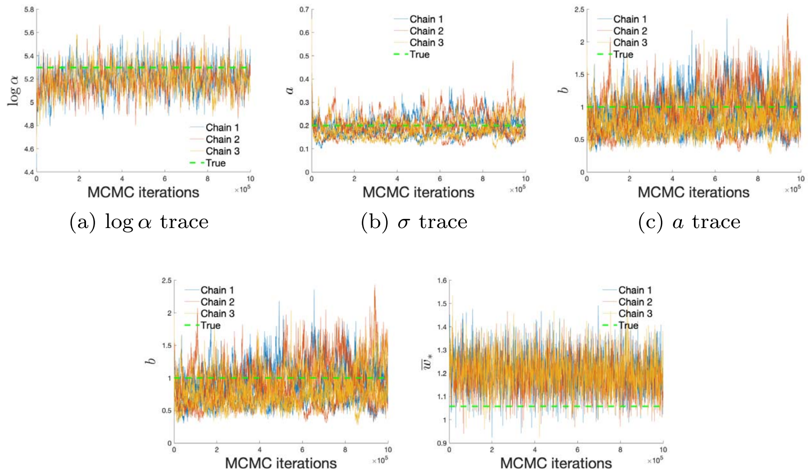
As before, to test convergence we calculate the approximate log-posterior probability density function (up to a normalizing constant). We see from Figure 30 that the chain has converged to the true value. This is confirmed by the Gelman-Rubin diagnostic, which in this case comes out as $\hat{R} = 1.03$. However, from the autocorrelation plot we see that the mixing is not as good as in the case without community structure, with dependencies not vanishing as quickly as we would like for increasing lag. This indicates that estimation in the setting with core-periphery and community structure can be more difficult, as we might expect.

D.1.3. No core-periphery structure

Here, we perform some additional experiments to test our model in two situations in which we do not expect it to work, in order to check that the behaviour is still sensible. Recalling that in our model the overall generated graph is still sparse, we first simulate data from the model of [6], with $\sigma > 0$. This model generates a sparse graph without a core-periphery structure. In this case, the size of the core estimated by our model tends to zero. Furthermore, we see from Figure 31(a) that we still accurately recover the degree distribution in this case.

Secondly, we test our model on a graph generated from the model of [6], but with $\sigma < 0$. In the construction of our model, the core and periphery are distinguished by the fact that the core nodes form a dense subgraph in an otherwise sparse graph. Thus, when the overall graph is dense, we do not have this distinction and our model struggles to identify any latent core-periphery structure. However, in this case this structure is not present. We see two different types of behaviour, depending on the parameters of the model and the initial conditions.

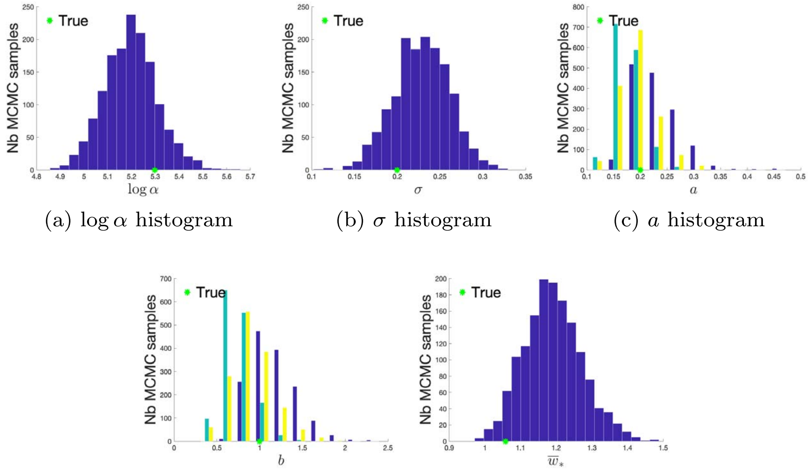
1. σ is (correctly) estimated to be negative, and the size of the core goes to zero. This is the behaviour we see in Figure 31(b) and is the behaviour we would expect.
2. σ is (incorrectly) estimated to be positive, and the whole graph is estimated to be in the core. Although this may not seem intuitive, it still fits with our definition of the core being a dense subgraph within a sparse network.



(a) $\log\alpha$ trace (b) σ trace (c) a trace

(d) b trace (e) $\bar{w}_*$ trace

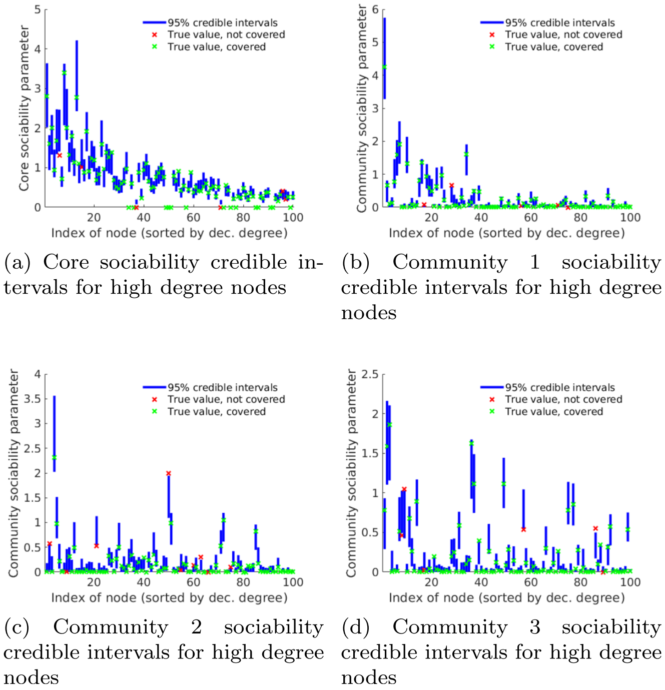
FIG 27. *MCMC trace plots parameters (a) $\log\alpha$, (b) σ, (c) a, (d) b and (e) $\bar{w}_*$. The graph is generated from the SparseCP model with $K = 4, \alpha = 200, \sigma = 0.2, \tau = 1, b_k = b = \frac{4}{K}, a_k = a = 0.2$. The green lines correspond to the values of the parameters used for generating the graph.*



(a) $\log\alpha$ histogram (b) σ histogram (c) a histogram

(d) b histogram (e) $\bar{w}_*$ histogram

FIG 28. *MCMC histograms for parameters (a) $\log\alpha$, (b) σ, (c) a, (d) b and (e) $\bar{w}_*$. The graph is generated from the SparseCP model with $K = 4, \alpha = 200, \sigma = 0.2, \tau = 1, b_k = b = \frac{4}{K}, a_k = a = 0.2$. The green stars correspond to the values of the parameters used for generating the graph.*



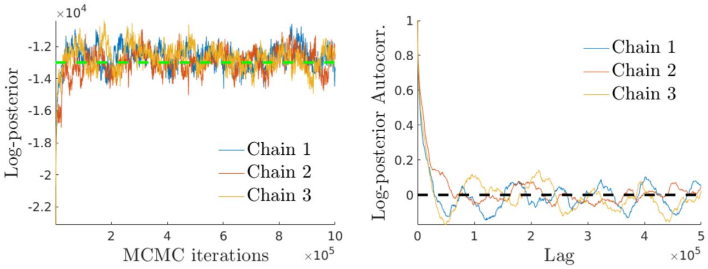
(a) Core sociability credible intervals for high degree nodes

(b) Community 1 sociability credible intervals for high degree nodes

(c) Community 2 sociability credible intervals for high degree nodes

(d) Community 3 sociability credible intervals for high degree nodes

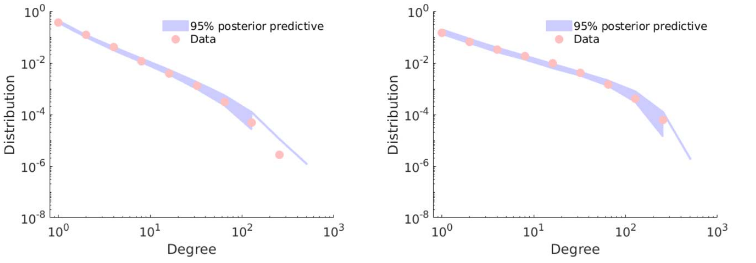
Fig 29. *Credible intervals for (a) the core sociability w_{i1}, (b) the community sociability w_{i2}, (c) the community sociability w_{i3}, (d) the community sociability w_{i4} of the 100 nodes with the highest degrees, for the graph generated with $K = 4, \alpha = 200, \sigma = 0.2, \tau = 1, b_k = b = \frac{4}{K}, a_k = a = 0.2$. In each case the true values are coloured green if they are covered by the credible interval, and red if not.*



(a) Log-posterior

(b) Autocorrelation

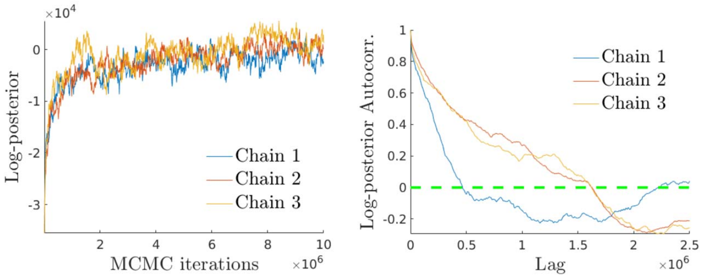
Fig 30. *Trace plot and autocorrelation of the log-posterior for a graph generated from the SparseCP model with $K = 4, \alpha = 200, \sigma = 0.2, \tau = 1, b_k = b = \frac{4}{K}, a_k = a = 0.2$. The green line in (a) corresponds to the value of the approximate log-posterior (up to a constant) under the model parameters used to generate the graph.*



(a) Posterior predictive degree distribution for a network with no core-periphery structure

(b) Posterior predictive degree distribution for a dense network

Fig 31. *Posterior predictive degree distributions fitting the SparseCP model to networks which (a) have no core-periphery structure, and (b) are dense. When the lower posterior predictive interval bound is 0, we plot only the upper interval bound.*



(a) Log-posterior

(b) Autocorrelation

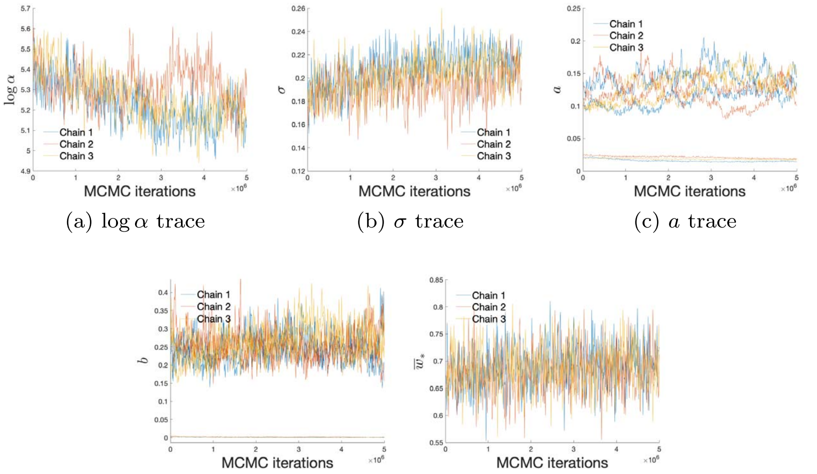
Fig 32. *Trace plot (left) and autocorrelation plot (right) of the approximated log-posterior probability density (up to a constant) of the US Airport network.*

D.2. Real data

D.2.1. US airport network

In Section 4.3.1, we present the results from performing posterior inference on the US Airport network. Here, we present some additional results specifically regarding the output of the MCMC sampler. In Figure 32 we give the trace plot and autocorrelation of the approximate logposterior. The Gelman-Rubin statistic comes out to be $\hat{R} = 1.25$ in this case, suggesting that the chain has not converged despite the large number of iterations. We can also calculate the Gelman-Rubin statistic for parameters α and σ. Here we obtain values of $\hat{R}_\alpha = 1.10$ and $\hat{R}_\sigma = 1.13$. Again, these are both above the threshold of 1.1 which we generally believe to indicate convergence. However, the lower values here indicate that the problem of convergence may not be as bad as indicated by the log-posterior.

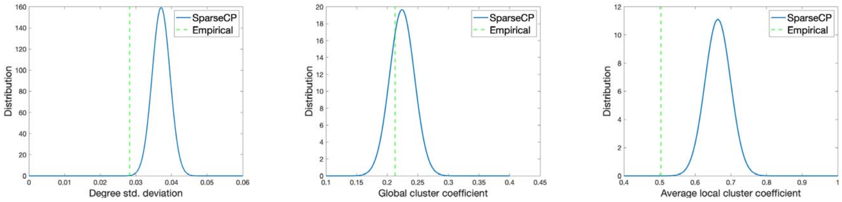
These results indicate that the convergence of our sampler is slow. As in



(a) $\log\alpha$ trace (b) σ trace (c) a trace

(d) b trace (e) $\bar{w}_*$ trace

Fig 33. *MCMC trace plots of parameters* (*a*) $\log\alpha$, (*b*) σ, (*c*) a, (*d*) b *and* (*e*) $\bar{w}_*$ *for the US Airport network.*



(a) Posterior predictive degree standard deviation

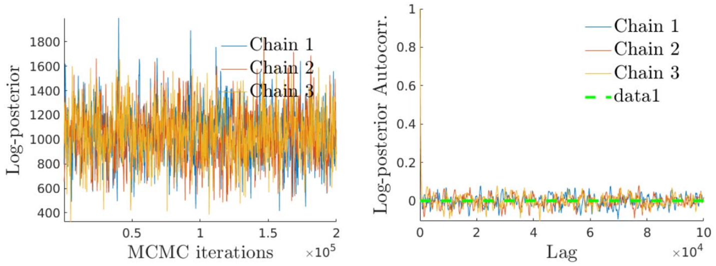
(b) Posterior predictive global cluster coefficient

(c) Posterior predictive average local cluster coefficient

Fig 34. *Posterior predictive plots for the US Airport network using the SparseCP model.* (*a*) *Standard deviation of degree;* (*b*) *global cluster coefficient (or triadic dependence);* (*c*) *average local cluster coefficient. Empirical values are given in green in each case.*

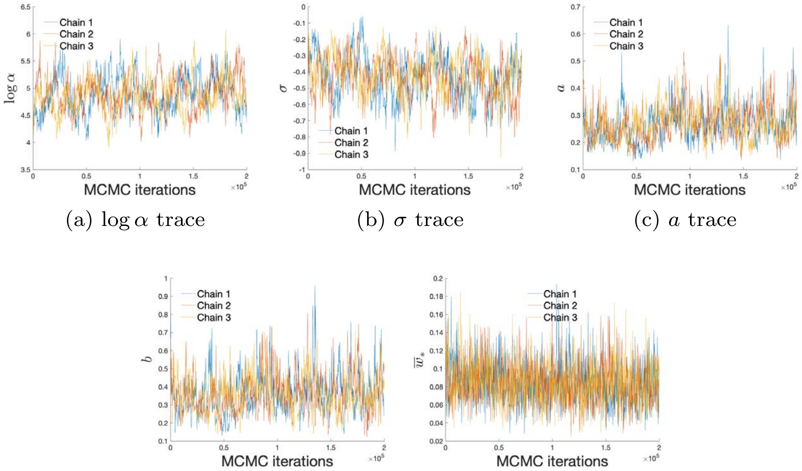
[41], even running a very long chain did not lead to an improvement. This long chain took roughly 60 hours to run, translating to a time cost of around 0.02 seconds per iteration. However, it is at least encouraging to see that the different chains are exploring parts of the parameter space with similar approximate log-posterior density. In Figure 33 we report the trace plots for the identifiable parameters of the US Airport network.

We can perform some posterior predictive checks, by comparing the posterior predictive distribution of the degree standard deviation, global cluster coefficient and average local cluster coefficient to the respective empirical values. In Figure 34 we see that the global cluster coefficient is being fit well, while the other statistics are being slightly overestimated.



(a) Log-posterior (b) Autocorrelation

FIG 35. *Trace plot (left) and autocorrelation plot (right) of the approximated log-posterior probability density (up to a constant) of the World Trade network.*



(a) $\log \alpha$ trace (b) σ trace (c) a trace

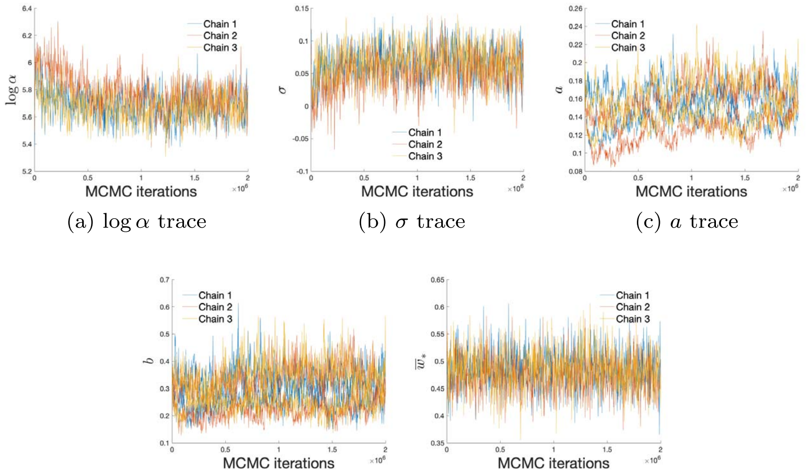
(d) b trace (e) $\bar{w}_*$ trace

FIG 36. *MCMC trace plots of parameters (a)* $\log \alpha$*, (b)* σ*, (c)* a*, (d)* b *and (e)* $\bar{w}_*$ *for the World Trade network.*

D.2.2. *World trade network*

In Section 4.3.2, we present the results from performing posterior inference on the World Trade network. Here, we present some additional results specifically regarding the output of the MCMC sampler. In Figure 35 we give the trace plot and autocorrelation of the approximate logposterior. In this case, the Gelman-Rubin statistic of $\hat{R} = 1.009$ suggests convergence of the Markov chain. As before, we can also calculate the Gelman-Rubin statistic for parameters α and σ. Here we obtain values of $\hat{R}_\alpha = 1.01$ and $\hat{R}_\sigma = 1.01$, providing further evidence of convergence.

In Figure 36 we see the trace plots for the identifiable parameters of the World Trade network.



(a) $\log\alpha$ trace

(b) σ trace

(c) a trace

(d) b trace

(e) $\bar{w}_*$ trace

Fig 37. *MCMC trace plots of parameters (a)* $\log\alpha$*, (b)* σ*, (c)* a*, (d)* b *and (e)* $\bar{w}_*$ *for the US Airport network without Alaskan airports.*

Appendix E: US airport network case study repeated without Alaskan airports

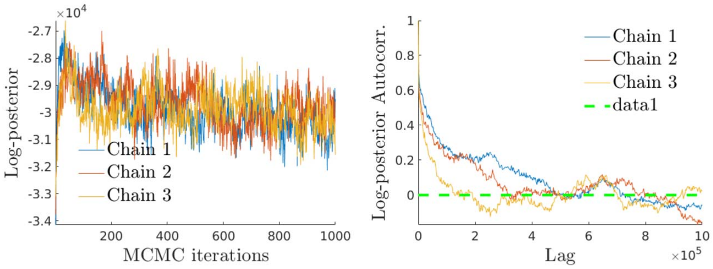
In Section 4.3.1, we note a problem occurring due to the sociability parameter for the Alaskan community. Here, we repeat our analysis having taken out all the Alaskan airports, as well as any airport only connected to Alaskan airports. The relevant plots comprise Figures 37-41.

In Figure 37 we see the trace plots for the identifiable parameters. From this we see that we no longer have the same problem of an estimate of one of the b_k being very close to 0.

In Figure 38 we see the autocorrelation plot of the approximate log-posterior density. This seems as though it is converging to a stable value. If we look at the Gelman-Rubin convergence diagnostic, this comes out to be $\hat{R} = 1.05$ in this case, indicating convergence. We also obtain values of $\hat{R}_\alpha = 1.06$ and $\hat{R}_\sigma = 1.03$, providing further evidence of convergence.

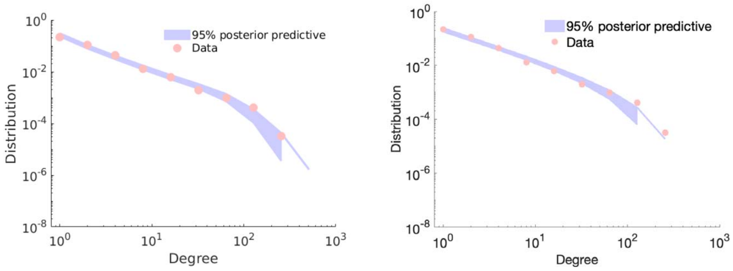
In Figure 39 we see the posterior predictive degree plot on the left, compared to the corresponding plot for the model of [41] with three communities on the right. We see that we are no longer overestimating the high degree nodes. We also see that the SNetOC model has estimated $\sigma < 0$ in this case, whereas for the SparseCP model we still estimate $\sigma > 0$.

If we look at the reweighted KS distances as before in Figure 40, we see that our model is still providing a better fit to the data, but not necessarily any better than the fit before excluding the Alaskan airports. In this case we report the Rényi statistics L_2 and U_1, because we see that our model is overestimating the number of low degree nodes, and underestimating the number of high



(a) Log-posterior (b) Autocorrelation

Fig 38. *Trace plot (left) and autocorrelation plot (right) of the approximated log-posterior probability density (up to a constant) of the US Airport network without Alaskan airports.*



(a) SparseCP posterior predictive degree distribution (b) SNetOC posterior predictive degree distribution

Fig 39. *Posterior Predictive Degree Distribution for the US Airport network without Alaskan airports, using (a) the SparseCP model and (b) the SNetOC model. When the lower posterior predictive interval bound is 0, we plot only the upper interval bound.*

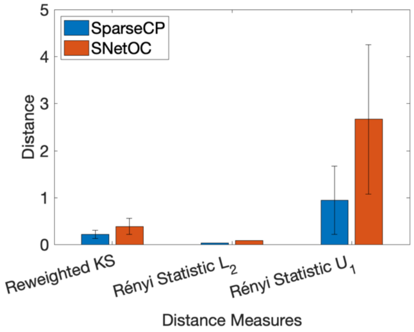


Fig 40. *Reweighted Kolmogorov Smirnov distance and Rényi statistics for assessing the degree distribution for the US Airport network without Alaskan airports.*

degree nodes.

Furthermore, we see that the core and communities identified here are largely the same as before (without the “Alaska” community). In Figure 41 we plot the

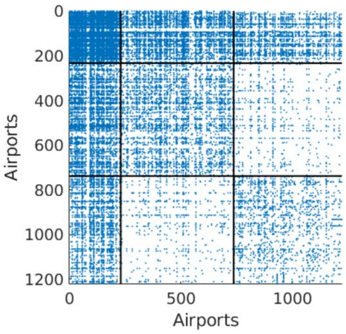


Fig 41. *Adjacency matrix for the US Airport network without Alaskan airports, ordered into core and periphery, and then by highest community weight, as estimated using the SparseCP model.*

adjacency matrix in this case, again ordered into core and periphery, and then by highest community weight.