Bayesian Methods for Statistical Network Analysis and Data Subsampling



Cian Vasanttilak Naik

St. Peter's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2022

Statement of Originality

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. My personal contributions are as outlined in the authorship forms at the end of each chapter. This dissertation is my own work except as specified in the text, acknowledgements, forms and papers.

Cian Vasanttilak Naik
Hilary 2022

Acknowledgements

I would like to first of all thank my supervisors Judith Rousseau and François Caron for their guidance. I am very grateful to have had such attentive and supportive supervisors, and this thesis would not have been possible without them.

I would also like to thank my unofficial supervisor Trevor Campbell for working with me remotely over the past year, and for all the effort he put in to help me get my final paper over the line in time.

I would next like to thank everyone else that I have worked with: Weisi Guo, Chenlei Leng, Cassidy Sugimoto, Vincent Larivière and Konstantina Palla, for their help and advice.

Thanks to everyone in the Oxford Statistics Department, especially the OxWaSP crew, for creating such a diverse and enjoyable setting to learn in.

I am grateful to the Engineering and Physical Sciences Research Council and Medical Research Council for providing me with the funding to undertake this opportunity.

Finally, I would like to thank my family, for their constant love and support, and all my friends, especially the Cave, for getting me through this with a smile on my face. I am very lucky to know them all.

Abstract

Bayesian statistics provides an important framework for parameter estimation and uncertainty quantification in a wide range of settings. Prior information on unknown parameters can be incorporated into Bayesian models in principled ways, resulting in posterior distributions that describe the uncertainty in those parameters. In this thesis we utilise this Bayesian framework to study problems in statistical network analysis and data subsampling. In both cases, the key task will be uncovering latent structure within datasets.

To perform statistical network analysis within the Bayesian setting, we use a Bayesian nonparametric (BNP) approach, in which infinitely many parameters are used to capture the complexities present in real-world data. This framework allows us to model sparse networks, a property commonly exhibited by real-world datasets. Furthermore, it allows us the flexibility to capture the structure exhibited by these networks, by modelling the popularity of nodes within them. We focus specifically on core-periphery networks, where there exists a subset of important, highly connected nodes within the overall network, and dynamic networks, where the structure of the network evolves over time. We perform posterior inference on these models using Markov chain Monte Carlo (MCMC).

One problem shared by our BNP network models and many other complex models is that exact inference is rarely possible, and thus approximate sampling methods such as MCMC are needed. However, these can be prohibitively slow for very large datasets. One solution to this problem is to replace the full dataset with a small, weighted subset - a *coreset* - that captures key features of the full dataset. The key idea here is that most of the data is often redundant, and an appropriately weighted subset can capture most of the relevant information. We introduce a novel Bayesian coreset construction algorithm, which is based on first uniformly subsampling the data, and then optimizing the weights. Our method is simple to implement, significantly faster than the state-of-the-art (in terms of performance), and comes with theoretical error guarantees.

Contents

1 Introduction **1**
1.1 Motivation 2
1.2 Statistical Network Analysis 6
1.3 Bayesian Nonparametric Network Models 10
1.4 Data Subsampling 18
1.5 Bayesian Coresets 19
1.6 Contributions 25

2 Sparse Networks with Core-Periphery Structure **29**

3 Bayesian Nonparametrics for Sparse Dynamic Networks **87**

4 Fast Bayesian Coresets via Subsampling and Quasi-Newton Refinement **115**

5 Conclusion **145**
5.1 Summary 145
5.2 Future Directions 146

Bibliography **149**

1 Introduction

Contents

1.1 Motivation 2
1.1.1 Statistical Network Analysis 2
1.1.2 Data Subsampling 4
1.2 Statistical Network Analysis 6
1.2.1 Sparsity 8
1.2.2 Power Law Degree Distributions 9
1.2.3 Exchangeability 9
1.3 Bayesian Nonparametric Network Models 10
1.3.1 Completely Random Measures 11
1.3.2 Properties 13
1.3.3 Generalised Gamma Process 14
1.3.4 Extensions 15
1.4 Data Subsampling 18
1.5 Bayesian Coresets 19
1.5.1 Coresets via Importance Sampling 20
1.5.2 Coresets via Sparse Regression 22
1.5.3 Coresets via Sparse Variational Inference 23
1.5.4 Other Bayesian Coreset Construction Methods 24
1.6 Contributions 25
1.6.1 Bayesian Nonparametric Network Models 26
1.6.2 Bayesian Coresets 27

This thesis is written in an integrated format, comprised of three publications within the field of Bayesian statistics. In this chapter, we begin by introducing the problems that we tackle, the motivation behind our work, and the themes that

connect them. We do this first in a non technical setting, before then introducing the statistical background necessary to present our work fully. The three publications that comprise the thesis are presented in Chapters 2 to 4. Finally, Chapter 5 concludes with a broad summary and some potential future research directions.

1.1 Motivation

1.1.1 Statistical Network Analysis

The first objective of this thesis is to understand and model networks. Network data arises in a wide range of fields, when considering the complex systems that comprise flight patterns, online interactions and trade relations, amongst others. Access to these types of data has become increasingly prominent in recent times, as have the computational tools that allow us to study large and complex networks.

A key component to understanding the behaviour of these systems is the identification of underlying structure within them. We first model a system as a network, with individual actors (such as people or airports) considered as nodes, and connections between them (such as friendships or flights) considered as edges. Then, we can study the patterns in how these edges are formed. In particular, we may want to understand the reasons why some nodes form far more connections than others, or investigate why there are higher numbers of edges between certain subgroups of nodes than others. A very common type of structure that we observe in the real world is community structure, where nodes within a given community have high numbers of edges between each other, but far fewer edges between them and nodes in other communities. These communities could correspond to friendship groups on social media, airports within certain geographical regions, or countries within the same trade bloc such as the European Union.

Another similar, but distinct, type of structure observed in real world networks is core-periphery structure. Here we observe two classes of nodes: *core* nodes that are highly connected amongst each other, and *periphery* nodes, which are more loosely connected to the core nodes, and sparsely connected to each other, if at all. We can see this type of structure within flight networks, with global hubs

such as Atlanta, Tokyo Haneda or London Heathrow having a large volume of flights, a large proportion of which are to other hub airports. Conversely, small regional airports have far fewer flights, and a high proportion of these may be to and from these larger hubs. In Figure 1.1 we see an example of a network with core-periphery structure, as identified by Jia and Benson (2019). Here, the nodes are airports, coloured in orange if they are core nodes, and blue otherwise, and the edges are flights. The data is from OpenFlights.[1]

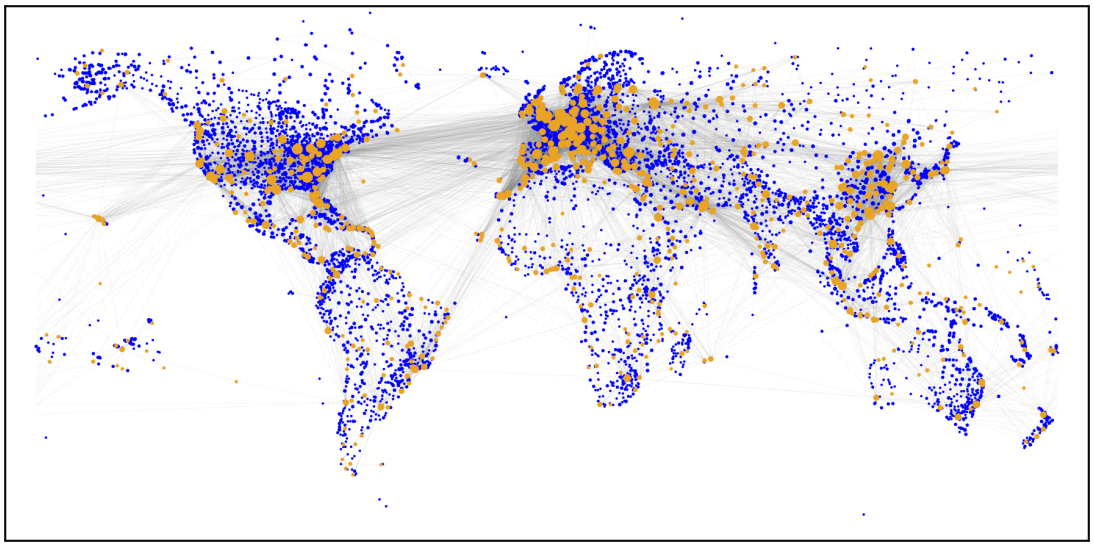


Figure 1.1: Vertex core scores from Jia and Benson (2019) for the OpenFlights global flight network. The top 10% of vertices with highest core scores are coloured orange, with the rest in blue. Vertex radius is proportional to square root of degree.

The first publication in this thesis is concerned with modelling networks with core-periphery structure. We want to be able to classify nodes into core or periphery, but also capture other important properties of the network. For example, the distribution of the number of connections (or degree) of each node. A central part of the strategy to do this is to assign a latent *sociability* parameter to each node, which measures how likely it is to make connections. In an online social network, this measure may indeed capture how sociable a person is. Accurately modelling these sociability parameters is the key to this work.

[1] `https://openflights.org/data.html#route`.

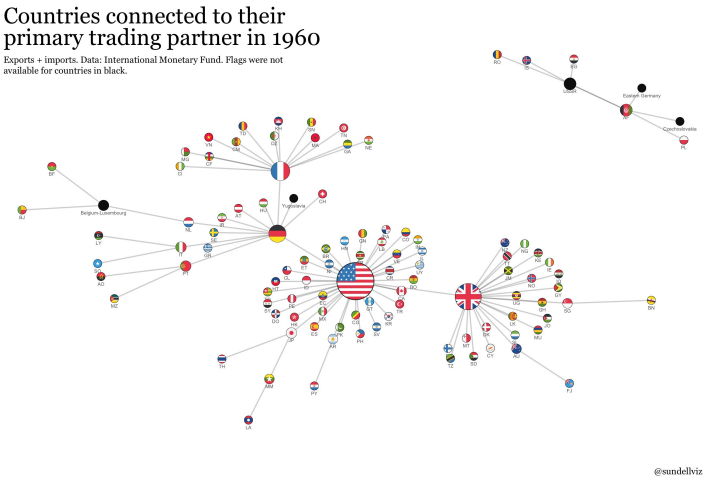
In reality, complex systems change and evolve over time. While thus far we have been concerned with static networks, the second publication in this thesis extends to the case of time-varying networks. An example of how drastically networks and their structure can change over time can be seen by looking at the network of world trading partners, where each node is a country and has exactly one edge connecting it to its biggest trading partner. We see in Figure 1.2 an analysis performed by Anders Sundell[2] with data from the International Monetary Fund (IMF).[3] Here we can clearly see the emergence of China and a European bloc of trade, where previously trade links had been centred around the US, UK and Germany.

Following the same framework as before, we model dynamic networks by assuming that node sociabilities can evolve over time. We then attempt to capture exactly how this evolution occurs. This allows us to model how the networks themselves can evolve more broadly. Our primary focus in both our pieces of work on network analysis is to understand the structure of the networks that we study. However, the complexity of the systems we study, as well as the methods used to model them, necessitates complicated inference techniques. The work on dynamic networks in particular uses approximate methods to simplify an otherwise intractable problem. Nevertheless, significant computation times are needed in order to fit our model to real world networks.

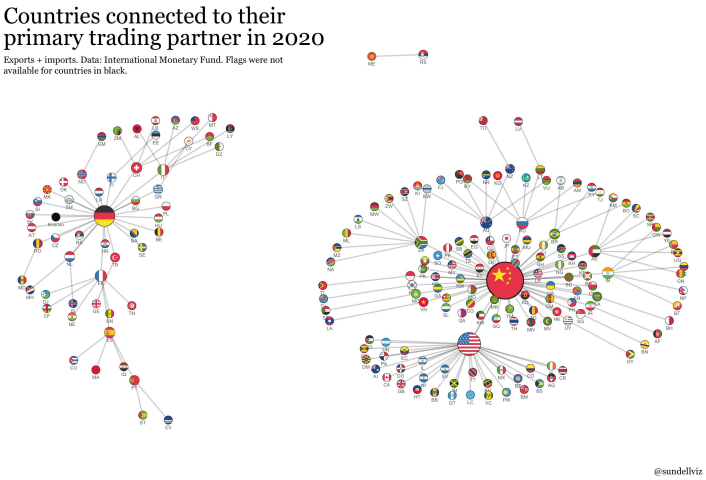
1.1.2 Data Subsampling

This brings us to the second objective of this thesis, which is to develop methods for reducing the computational cost of performing inference in large models. Whilst this problem has been widely studied, the technique we focus on relies on the idea of *redundancy*. The idea is that, in a large dataset, a small subset of the data may be sufficient to describe the whole in some sense, with the rest of the dataset being redundant. By weighting that subset appropriately, we can therefore use

[2] `https://twitter.com/sundellviz/status/1488596964208001026`.
[3] `https://data.imf.org/regular.aspx?key=61013712`.
[4] See footnote 2.



(a) Trade partners in 1960



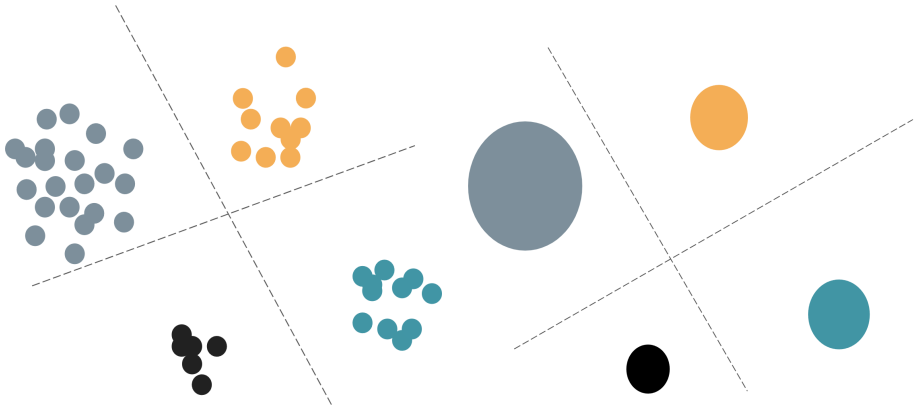
(b) Trade partners in 2020

Figure 1.2: Networks of countries connected to their main trading partner in (a) 1960 and (b) 2020, from Anders Sundell.[4]Nodes are coloured using the country flag, or in black where flags where not available.

the smaller weighted subset of data in place of the whole dataset, at significantly reduced computational cost.

An example of how this could work is shown in Figure 1.3. Here, we can imagine we are trying to estimate the decision boundary when performing a clustering problem. The weighted data shown on the right hand plot can be used to obtain a decision boundary that is a good approximation of that found using the whole dataset (as in the left hand plot). Here, the redundancy is between the points within each cluster. In practice, all we may need is one point located at the cluster

centre, with a weight corresponding to the size of the cluster. This example is deliberately simplistic, but gives an idea of the kind of problems we can hope to speed up using subsampling.



(a) Full clustering **(b)** Coreset clustering

Figure 1.3: Example of estimating the decision boundary in a toy clustering problem, using (a) the full data, and (b) a weighted subset of the data.

The third publication in the thesis is concerned with developing a method to select the weighted subset of data discussed above, called a *coreset*. Our approach aims to address some of the practical and theoretical problems shared by previous methods. In particular, for a subsampling method to be practically useful, we first need to be able to construct the coreset in a reasonable amount of time and space. Otherwise, there is nothing gained versus simply performing inference on the full dataset. Secondly, we want some guarantees on the error we are introducing by using a coreset of the data. Guaranteeing this low error rules out naive approaches such as simply uniformly sampling the data, but also more recent methods.

1.2 Statistical Network Analysis

We now introduce the technical background necessary to tackle the problems discussed in the previous section. We start with the field of statistical network analysis. Specifically, we focus on the design of random graph models, which originated with the work of Erdös and Rényi (1959). Commonly referred to as the Erdős-Rényi model, a graph G with n nodes, or vertices, is constructed by

including each edge between two nodes in the graph with probability p. We denote this graph model as $G(n, p)$.

A great deal of work has gone into building upon this work to construct models for networks that incorporate certain salient properties. For example, an important class of models is based on graphons (Wolfe and Olhede, 2013; Wolfe and Choi, 2014; Gao et al., 2015; Orbanz and Roy, 2015; Zhang et al., 2017). Graphons are symmetric measurable functions $W : [0, 1]^2 \to [0, 1]$ that can be used to define a random graph model as follows. Firstly, each vertex j is assigned to an independent random number $u_j \sim U[0, 1]$ drawn from a standard uniform distribution. Then, edges are formed by including each edge (i, j) independently, with probability given by $W(u_i, u_j)$.

A special case of these graphon models are those based on the stochastic block model (Holland et al., 1983; Nowicki and Snijders, 2001; Snijders and Nowicki, 1997). Here, vertices are partitioned into disjoint subsets $C_1, \ldots, C_r$ called communities. Edges are then formed at random, with an edge (i, j) forming between two vertices $i \in C_u$ and $j \in C_v$ with probability P_{uv}, where P is a symmetric $r \times r$ matrix of probabilities.

There are many more models that we do not discuss here, though a review of some of these can be found in the book Newman (2018). In the rest of this section we restrict ourselves to those which will be relevant for our work. Throughout, we will focus on undirected graphs, where edges between nodes are bidirectional, and thus the ordering of the nodes that comprise an edge is unimportant. Before we start, we also need to introduce some notation.

Notation. We follow the asymptotic notation of Janson (2011). 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. $\iff \limsup_{\alpha \to \infty} \left| \frac{X_\alpha}{Y_\alpha} \right| < \infty$ a.s.; $X_\alpha = o(Y_\alpha)$ a.s. $\iff \limsup_{\alpha \to \infty} \left| \frac{X_\alpha}{Y_\alpha} \right| = 0$ a.s.; $X_\alpha = \Theta(Y_\alpha)$ a.s. $\iff$ $X_\alpha = O(Y_\alpha)$ and $Y_\alpha = O(X_\alpha)$ a.s.

1.2.1 Sparsity

The first relevant property we introduce is the idea of a sparse graph. The reason for this focus on sparse graphs is that many real world networks are believed to be sparse (Newman, 2009; Barabási et al., 2016; Orbanz and Roy, 2015). When talking about a finite graph G this term is sometimes used informally to refer to graphs in which most of the nodes are unconnected. Conversely, a graph is said to be dense if most or all of the nodes are connected. However, to introduce the definition of sparsity that we will use in our work, we need to take an asymptotic view. We say that a graph is sparse if the number of edges scales subquadratically with the number of nodes, using the terminology of Bollobás and Riordan (2009).

To make this precise, let $G = (G_\alpha)_{\alpha \geq 0}$ be a family of growing undirected random graphs with no isolated nodes, where $\alpha \geq 0$ can be thought of as a size parameter. Further, specify that $G_\alpha = (V_\alpha, E_\alpha)$, where V_α and E_α are the set of nodes 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 $\alpha \to \infty$.

This notation allows us to give the definition of sparsity for the family $G = (G_\alpha)_{\alpha \geq 0}$.

Definition 1.2.1 (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

$$N_\alpha^{(e)} = \Theta(N_\alpha^2) \tag{1.2.1}$$

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

$$N_\alpha^{(e)} = o(N_\alpha^2) \tag{1.2.2}$$

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

1.2.2 Power Law Degree Distributions

In a sparse graph, most nodes may be unconnected with each other. Of particular interest for us is the setting where most nodes have only a small number of connections, but some have a far more. One situation where this occurs is when a network has a power law degree distribution. By this, we mean that the proportion $P(k)$ of nodes in the graph that have k connections to other nodes behaves like $P(k) \sim k^{-\gamma}$ for large values of k (where typically, $2 < \gamma < 3$). Many real-world networks have been reported to have this property (Clauset et al., 2009), though just how prevalent they are has been debated in recent times (Broido and Clauset, 2019; Holme, 2019). Nevertheless, they remain an interesting type of network to study.

One example of such a network can be generated using the Barabási-Albert model (Albert and Barabási, 2002) as follows. Firstly, the network is initiated with a fully connected set of m_0 nodes. Nodes are then sequentially added to the network, by connecting them to $m \leq m_0$ existing nodes. The probability of connecting a new node to node i is given by

$$p_i = \frac{k_i}{\sum_j k_j},$$

where k_i is the degree of node i. This way, the degrees of initially well-connected nodes increase disproportionately, leading to the power law degree distribution detailed above. In fact, Albert and Barabási (2002) show that the model results in a network with a power law degree distribution, with exponent $\gamma = 3$.

1.2.3 Exchangeability

Another important property to discuss is the idea of exchangeability. This concept arises from the fact that when we study a network we may want to assume that the order in which the nodes are observed does not matter (Bickel and Chen, 2009; Hoff, 2009). In order to define exchangeability, we also introduce some more notation. A graph G can be represented by a binary adjacency matrix Z, where $Z_{ij} = 1$ if there is an edge from node i to node j. For undirected graphs, we also have that

$Z_{ji} = Z_{ij}$. Then, for any matrix Z, an (infinite) jointly exchangeable random array (Diaconis and Janson, 2008; Lauritzen, 2008) is defined by the condition that

$$(Z_{ij}) \stackrel{d}{=} \left(Z_{\pi(i)\pi(j)}\right) \text{ for } (i,j) \in \mathbb{N}^2 \tag{1.2.3}$$

for any permutation π of $\mathbb{N}$, where by $\stackrel{d}{=}$ we mean that these two random arrays have the same distribution.

The Aldous-Hoover theorem (Aldous, 1981; Hoover, 1979) for exchangeable arrays has lead to the development of many Bayesian statistical network models (Lloyd et al., 2012; Hoff et al., 2002; Nowicki and Snijders, 2001; Airoldi et al., 2008; Kim and Leskovec, 2012; Miller et al., 2009; Bickel and Chen, 2009; Bickel et al., 2011; Rohe et al., 2011; Zhao et al., 2012; Airoldi et al., 2014; Wolfe and Choi, 2014). However, a major drawback exposed by the theorem is that graphs that are represented by an exchangeable array are either empty or dense (Orbanz and Roy, 2015). Graphon models fall into this category.

This tension motivates the development of the Bayesian nonparametric framework for statistical network modelling introduced in Caron and Fox (2017). We devote the next section to discussing this, and we will see that it enables the study of graphs which are both sparse (with power-law degree distribution), and exhibit (a different form of) exchangeability.

1.3 Bayesian Nonparametric Network Models

Bayesian nonparametrics is a branch of Bayesian statistics in which parameters can be infinite dimensional. In this thesis, we only review some central concepts needed for statistical network analysis, but a more thorough treatment can be found in the books (Ghosal and Van der Vaart, 2017; Hjort et al., 2010; Daley and Vere-Jones, 2008). The use of infinitely many parameters means that the complexity of models can adapt to data in question. In the context of network modelling, it provides a natural setting for the study of sparse graphs. This is because it allows for the consideration of graphs that may have infinitely many nodes, only finitely many of which form connections.

To this end, instead of representing a graph G by a finite dimensional adjacency matrix, Caron and Fox (2017) instead represent it by the following infinite point process on the plane $\mathbb{R}_+^2 = [0, \infty)^2$

$$Z = \sum_{i,j} z_{i,j} \delta_{(\theta_i, \theta_j)} \tag{1.3.1}$$

where $z_{i,j} = z_{j,i} = 1$ if i and j are connected, and 0 otherwise. Each node i is located at some point $\theta_i \in \mathbb{R}_+ = [0, \infty)$. A finite graph G_α of size $\alpha > 0$ can be obtained by restricting Z to $[0, \alpha]^2$, and we refer to the restricted point process as Z_α. α can be thought of as representing time, and the properties of the graph as α grows relate to how new potential nodes and resulting edges enter the network.

A generative model is used to construct Z, where nodes i and j are connected with probability

$$Pr(z_{i,j} = 1 | (w_l)_{l=1,2,\ldots}) = \begin{cases} 1 - e^{-2w_i w_j} & i \neq j \\ 1 - e^{-w_i^2} & i = j \end{cases} \tag{1.3.2}$$

where the $w_i > 0$ can be thought of a sociability parameters for each node i. Modelling the latent structure introduced by defining these node-level parameters will be a key part of our work within this framework.

1.3.1 Completely Random Measures

The sociability parameters $w_i > 0$ and node indices $\theta_i \in \mathbb{R}_+$ are specified jointly as $(w_i, \theta_i)_{i=1,2,\ldots}$ by defining the atomic measure

$$W = \sum_{i=1}^{\infty} w_i \delta_{\theta_i} \tag{1.3.3}$$

where W is distributed according to a completely random measure (CRM) (Kingman, 1967; Daley and Vere-Jones, 2008; Lijoi and Prünster, 2010). This is a central concept in this framework.

Definition 1.3.1. A completely random measure (CRM) W on $\mathbb{R}_+$ is a random measure such that, for all finite families of disjoint, bounded measurable sets $A_1, \ldots, A_n$ of $\mathbb{R}_+$, the random variables $W(A_1), \ldots, W(A_n)$ are mutually independent.

Kingman (1967) shows that a CRM can be decomposed into a sum of three independent components:

$$W = W_0 + W_f + W_r. \tag{1.3.4}$$

where W_0 is a non-random measure, and W_f is given by

$$W_f = \sum_{j=1}^{J} u_j \delta_{x_j}$$

where J is constant, but can be an integer or infinity. The x_j are fixed and called the fixed atoms of W, and the u_j are mutually independent random variables on $\mathbb{R}_+$. W_r is given by

$$W_r = \sum_{i=1}^{\infty} w_i \delta_{\theta_i}$$

where $(w_i, \theta_i)_{i=1,2,\ldots}$ are the points of a Poisson process on $(0, \infty) \times \mathbb{R}_+$ with mean measure $\nu(dw, d\theta)$. In the random graph framework, only CRMs with no deterministic component or fixed atoms are considered, so that both W_0 and W_f are equal to 0. In this case W takes the form given by (1.3.3).

Furthermore, only homogeneous CRMs with stationary increments are considered, meaning that the locations θ_i are independent of the weights w_i, and the mean measure can be decomposed as $\nu(dw, d\theta) = \rho(w)dw\lambda(d\theta)$, where λ is the Lebesgue measure and ρ is a Lévy density on $(0, \infty)$ such that

$$\int_0^{\infty} (1 - e^{-w})\rho(w)dw < \infty$$

We denote this by

$$W \sim CRM(\rho, \lambda). \tag{1.3.5}$$

If

$$\int_0^{\infty} \rho(w)dw = \infty$$

then there will be almost surely an infinite number of jumps in any interval $[0, \alpha]$ and the CRM is called infinite activity. Otherwise, it is called finite activity.

1.3.2 Properties

We now review the relevant properties of networks we introduced above, in the context of the framework of Caron and Fox (2017). The first of these relates to the sparsity of the model.

1.3.2.1 Sparsity

To investigate the sparsity properties of the graph model, we consider the restricted graph Z_α as $\alpha \to \infty$. The nodes and edges in the restricted graph are respectively given by

$$N_\alpha = \sum_i \mathbb{1}_{\theta_i \leq \alpha} \mathbb{1}_{(\sum_j z_{ij} \mathbb{1}_{\theta_j \leq \alpha}) \geq 1} \tag{1.3.6}$$

which is the number of nodes with degree at least one in the network, and

$$N_\alpha^{(e)} = \sum_{i \leq j} z_{ij} \mathbb{1}_{\theta_i \leq \alpha} \mathbb{1}_{\theta_j \leq \alpha} \tag{1.3.7}$$

which is the number of edges in the network. The sparsity properties of the model are characterised in the following theorem

Theorem 1.3.1. *Consider the point process Z as defined in (1.3.1) with $\rho(w) \neq 0$. Let $N_\alpha^{(e)}$ be the number of edges in the undirected graph restriction Z_α, and N_α be the number of nodes. If the CRM W is finite-activity:*

$$\int_0^\infty \rho(w) dw < \infty$$

then the number of edges scales quadratically with the number of nodes

$$N_\alpha^{(e)} = \Theta\left(N_\alpha^2\right)$$

*almost surely as α tends to infinity, and the graph is **dense**. If the CRM is infinite-activity, i.e.*

$$\int_0^\infty \rho(w) dw = \infty$$

and we also have that

$$\int_0^\infty w\rho(w) dw < \infty$$

then the number of edges scales sub-quadratically

$$N_\alpha^{(e)} = o\left(N_\alpha^2\right)$$

*almost surely as α tends to infinity, and the graph is **sparse**.*

Thus, with the correct choice of $\rho(w)$, we can construct sparse graphs using this model. Moreover, it turns out that if we consider a particular notion of exchangeability, then we can construct graphs that are both sparse and exchangeable.

1.3.2.2 Exchangeability

The idea of exchangeability in this framework can be thought of as follows. A point process Z on $\mathbb{R}_+^2$ is jointly exchangeable if and only if, for any $h > 0$ and for any permutation π of $\mathbb{N}$,

$$(Z(A_i \times A_j)) \stackrel{d}{=} \left(Z\left(A_{\pi(i)} \times A_{\pi(j)}\right)\right) \text{ for } (i,j) \in \mathbb{N}^2,$$

where here we consider intervals $A_i = [h(i-1), hi]$ with $i \in \mathbb{N}$. An equivalent representation theorem to that of (Aldous, 1981; Hoover, 1979) for such jointly exchangeable point processes is given by Kallenberg (2005). Under that framework, we have the following result.

Theorem 1.3.2. *For any CRM $W \sim CRM(\rho, \lambda)$, the point process Z defined by (1.3.1)-(1.3.3) is jointly exchangeable.*

1.3.3 Generalised Gamma Process

In practice, the choice of $\rho(w)$ used by Caron and Fox (2017), and the one that we use in our work, is the Lévy intensity of a generalised gamma process (GGP) (Hougaard, 1986; Aalen, 1992; Lee and Whitmore, 1993; Brix, 1999), which is a CRM with two interpretable parameters and useful conjugacy properties (Lijoi et al., 2007; Caron et al., 2014). The Lévy intensity of the GGP is given by

$$\rho(w) = \frac{1}{\Gamma(1-\sigma)} w^{-1-\sigma} \exp(-\tau w)$$

where the two parameters (σ, τ) satisfy

$$(\sigma, \tau) \in (-\infty, 0] \times (0, +\infty) \text{ or } (\sigma, \tau) \in (0, 1) \times [0, +\infty).$$

We refer to the resulting CRM, restricted to $[0, \alpha]$, as $W \sim GG(\alpha, \sigma, \tau)$. If $\sigma < 0$, the GGP is a finite-activity CRM, and the resulting graph is dense. If $\sigma \geq 0$, then the GGP is infinite activity, and the resulting graph is sparse. Moreover, Caron et al. (2017) show that if $\sigma \in (0, 1)$ the graph has a power law degree distribution. Specifically, if we denote by $N_{\alpha,j}$ the number of nodes with degree j, then for all $j \geq 1$,

$$\frac{N_{\alpha,j}}{N_\alpha} \xrightarrow{\alpha \uparrow \infty} p_{\sigma,j} = \frac{\sigma \Gamma(j - \sigma)}{\Gamma(1 - \sigma)\Gamma(j + 1)},$$

almost surely. In particular, for large j, we have the tail behaviour

$$p_{\sigma,j} \overset{j \uparrow \infty}{\sim} \frac{\sigma}{\Gamma(1 - \sigma)} j^{-1-\sigma},$$

which corresponds to a power law behaviour with exponent $\gamma = 1 + \sigma$. Note that in this case $\gamma \in (1, 2)$.

1.3.4 Extensions

We conclude this section by introducing some extensions to the framework, and introducing how our work fits in to it.

1.3.4.1 Overlapping Community Structure

Todeschini et al. (2020) extend the framework to model graphs with overlapping community structure. As before, each node i is located at some point $\theta_i \in \mathbb{R}_+$, and is assigned to a set of latent non-negative parameters w_{ik}, $k = 1, \ldots, p$. The probability of a connection between nodes i and j is then given by

$$Pr(z_{i,j} = 1 | (w_{l1}, \ldots, w_{lp})_{l=1,2,\ldots}) = \begin{cases} 1 - e^{-2\sum_{k=1}^{p} w_{ik} w_{jk}} & i \neq j \\ 1 - e^{-\sum_{k=1}^{p} w_{ik}^2} & i = j \end{cases} \tag{1.3.8}$$

The model parameters $(w_{i1}, \ldots, w_{ip}, \theta_i)_{i=1,2,\ldots}$ can be described using the concept of a vector of CRMs (Epifani and Lijoi, 2010; Leisen and Lijoi, 2011; Leisen et al.,

2013; Griffin et al., 2013; Lijoi et al., 2014; Griffin and Leisen, 2017). More specifcally, we can define a vector of CRMs $(W_1, \ldots, W_p)$ on $\mathbb{R}_+$:

$$W_k = \sum_{i=1}^{\infty} w_{ik}\delta_{\theta_i}, \qquad k = 1, \ldots, p$$

where $(w_{i1}, \ldots, w_{ip}, \theta_i)_{i=1,2,\ldots}$ are the points of a Poisson process on $(0, \infty)^p \times \mathbb{R}_+$ with mean measure $\rho(dw_1, \ldots, dw_p)\lambda(d\theta)$. We write

$$(W_1, \ldots, W_p) \sim CRM(\rho, \lambda). \tag{1.3.9}$$

1.3.4.2 Core-periphery Structure

The concentration of ρ on $(0, \infty)^p$ in the overlapping communities construction of Todeschini et al. (2020) means that the sparsity properties are homogeneous across the whole network. We may instead want to model networks with a dense *core* of nodes, within an overall sparse graph. This so-called core-periphery structure appears in many real-world networks (Holme, 2005; Csermely et al., 2013), and a number of approaches have been proposed for its detection, see Rombach et al. (2017) for a review.

In Chapter 2, we present a model for sparse graphs with both community and core-periphery structure, based on an extension to the framework of Caron and Fox (2017). The key difference in our formulation is that we introduce a *core* parameter that can be equal to zero. We can partition the network based on which nodes have a core parameter equal to zero, and which do not. Carefully choosing the distribution of this parameter allows us to obtain the desired heterogeneity in sparsity properties across the network.

1.3.4.3 Dynamic Networks

In Chapter 3, we present an extension of the above framework to dynamically evolving networks. Again, each node i is located at some point $\theta_i \in \mathbb{R}_+$, but now is assigned to a non-negative sociability parameter w_{ti} for each time $t = 1, \ldots, T$. The probability of a connection between nodes i and j at time t is given by

$$\Pr(z_{tij} = 1 \mid (w_{t-k,i}, w_{t-k,j})_{k=0,\ldots,t-1}) = \begin{cases} 1 - e^{-2w_{ti}w_{tj}} & i \neq j \\ 1 - e^{-w_{ti}^2} & i = j \end{cases} \tag{1.3.10}$$

We then consider the associated collection of random measures on $\mathbb{R}_+$, for $t = 1, \ldots, T$

$$W_t = \sum_i w_{ti}\delta_{\theta_i}, \quad t = 1, \ldots, T.$$

The key part of our work is carefully choosing the distribution of the sequence of random measures $(W_t)_{t=1,2,\ldots,T}$. In particular we introduce dependence between the random measures at different time points, so that we may model how the network evolves over time, rather than just considering the networks to be independent from one time point to the next.

1.3.4.4 BFRY Approximation

In the models of (Caron and Fox, 2017; Todeschini et al., 2020) and our own work on core-periphery networks, the likelihood is quite complex. However, it is still possible to perform inference using a carefully constructed MCMC algorithm. However, when moving to the dynamic network setting, this becomes intractable. Thus, in Chapter 3, we instead consider an approximate inference method, using a finite-dimensional approximation to the GGP introduced by Lee et al. (2016). This approximation gives rise to a particularly simple conjugate construction, enabling posterior inference to be performed.

Let $\text{BFRY}(\eta, \tau, \sigma)$ denote a (scaled and exponentially tilted) BFRY random variable on $(0, \infty)$, as introduced by (Bertoin et al., 2006; Devroye and James, 2014; Lee et al., 2016) and named after the authors Bertoin, Fujita, Roynette and Yor of the first of those articles. The probability density function is given by

$$g_{\eta,\tau,\sigma}(w) = \frac{\sigma w^{-1-\sigma} e^{-\tau w}\left(1 - e^{-(\sigma/\eta)^{1/\sigma} w}\right)}{\Gamma(1-\sigma)\left\{(\tau + (\sigma/\eta)^{1/\sigma})^{\sigma} - \tau^{\sigma}\right\}}$$

with parameters $\sigma \in (0, 1)$, $\tau > 0$ and $\eta > 0$.

We then consider the finite-dimensional measure

$$W' = \sum_{i=1}^{K} w_{1i}\delta_{\theta_i}$$

where $w_{1i} \sim \text{BFRY}(\alpha/K, \tau, \sigma)$ and $K < \infty$ is the truncation level. As shown by Lee et al. (2016), for $\sigma \in (0, 1)$

$$W' \xrightarrow{d} GG(\alpha, \sigma, \tau)$$

as the truncation level K tends to infinity. By this we mean that, if $W \sim GG(\alpha, \sigma, \tau)$ is the GGP CRM used by Caron and Fox (2017), then

$$\lim_{K \to \infty} \mathcal{L}_f(W') = \mathcal{L}_f(W)$$

for an arbitrary measurable and positive f, where

$$\mathcal{L}_f(W) \coloneqq \mathbb{E}\left[e^{-W(f)}\right]$$

is the Laplace functional of W as defined by Lee et al. (2016). The details of how this approximation allows us to model dynamic networks are given in Chapter 3.

1.4 Data Subsampling

In many Bayesian methods, exact inference is impossible due to the complexity of the models. Thus, approximate inference schemes such as Markov chain Monte Carlo (MCMC) are needed (Robert and Casella, 2004; Robert and Casella, 2011; Gelman et al., 2014, Ch. 11,12). Indeed, in the Bayesian nonparametric network models within the framework of Caron and Fox (2017) this is what we utilise. However, in the large data setting MCMC can run into computational problems (Bardenet et al., 2017), as they generally have a time complexity that at best scales linearly with the dataset size N. This can quickly become intractable when N is large.

Many methods have been proposed to deal with this problem. We focus here on subsampling, which is motivated by stochastic methods in variational inference (Hoffman et al., 2013; Ranganath et al., 2014). The cost of MCMC can be reduced by only including a random subsample of $M \ll N$ data points in each Markov chain iteration (Bardenet et al., 2017; Korattikara et al., 2014; Maclaurin and Adams, 2014; Welling and Teh, 2011; Ahn et al., 2012; Quiroz et al., 2018). However, this can give rise to substantial error in the stationary distribution of the resulting Markov chain.

An alternative approach attempts to performing subsampling in a more principled way by understanding the latent structure of the data. In particular, the key insight is that most of the data in a dataset is often redundant, but there may be a small subset of the data which is representative of the whole in some sense, and thus can be used in its place. This idea can be readily applied to the problem of MCMC (or indeed any other approximate Bayesian inference procedure) as a preprocessing step. First, a small subset of $M \ll N$ data points are selected, and then whichever inference scheme was previously being utilised can be run on the smaller dataset, at a far reduced computational cost.

Many methods exists to subsample data, from simple uniform subsampling approaches (Politis et al., 1999) that share their roots with the jackknife (Quenouille, 1956; Tukey, 1958) and bootstrap (Efron, 1992; Efron and Tibshirani, 1994) resampling methods to importance sampling (see Tokdar and Kass (2010) for a review). Following our intuition that we need to select a subset that approximates the whole dataset in some sense, we focus on methods for constructing a *coreset*. A coreset is a small, weighted subset of data, the concept of which first originated in the computational geometry literature (Agarwal et al., 2005; Feldman and Langberg, 2011). The idea has then been developed further in the frequentist literature, especially in the settings of clustering (Bachem et al., 2015, 2016; Feldman et al., 2020; Lucic et al., 2016), and mixture models (Feldman et al., 2011; Lucic et al., 2017*a*,*b*). As the focus of our work is in the Bayesian setting, we specifically consider the construction of *Bayesian coresets*.

1.5 Bayesian Coresets

As introduced by Huggins et al. (2016), the role of coresets in the Bayesian setting is to approximate a posterior distribution of interest with a coreset posterior approximation. To make this idea precise, we consider the following setting.

Suppose we want to sample from a target posterior distribution $\pi(\theta)$ for variable $\theta \in \Theta$ that is comprised of log-likelihood $\mathcal{L}(\theta) \coloneqq \sum_{n=1}^{N} f_n(\theta)$ and prior $\pi_0(\theta)$,

$$\pi(\theta) \coloneqq \frac{1}{Z} \exp\left(\sum_{n=1}^{N} f_n(\theta)\right) \pi_0(\theta), \tag{1.5.1}$$

where Z is the normalizing constant. A typical setting is when we have N data points X_n which are i.i.d. conditioned on θ. In this case, $f_n(\theta)$ can be written as $f_n(\theta) = \log p(X_n|\theta)$. Since obtaining samples approximately distributed from this posterior is expensive when N is large, we instead look for a set of weights $w \in \mathbb{R}_+^N$ corresponding to each data point, where at most $M \ll N$ of these are nonzero, i.e. $\|w\|_0 \coloneqq \sum_{n=1}^{N} \mathbb{1}_{w_n>0} \leq M$. The coreset corresponds to the subset of data points with strictly positive weights. We can then approximate the full posterior using the coreset posterior given by

$$\pi_w(\theta) \coloneqq \frac{1}{Z(w)} \exp\left(\sum_{n=1}^{N} w_n f_n(\theta)\right) \pi_0(\theta), \tag{1.5.2}$$

where $Z(w)$ is the new normalizing constant. Under this notation, we use $\pi_1 = \pi$ to denote the coreset posterior with $w_n = 1\ \forall n$, which corresponds to the full posterior. Sampling from this posterior is much quicker if $M \ll N$, as it has a complexity that will typically scale linearly in M, rather than N.

The key challenge is of course to construct the weights w so that π_w is close to π_1, while retaining the sparsity property which gives the computation speedup. An important part of this is to define what we mean by "close". Several different discrepancy measures have been used to measure how close π_1 and π_w are, and this choice can suggest a coreset construction method. We review the main methods here.

1.5.1 Coresets via Importance Sampling

The first Bayesian coreset approach, introduced by Huggins et al. (2016), constructs a coreset by considering "closeness" under the uniform norm, weighted by the inverse of the total log-likelihood $\mathcal{L}(\theta)$

$$\|g(\theta)\|_\infty \coloneqq \sup_{\theta\in\Theta} \left|\frac{g(\theta)}{\mathcal{L}(\theta)}\right|. \tag{1.5.3}$$

Using this formulation, the coreset construction problem can be written as:

$$\min_{w \in \mathbb{R}^N} \|\mathcal{L}(w, \theta) - \mathcal{L}(\theta)\|_\infty^2 \qquad \text{s.t. } w \geq 0, \|w\|_0 \leq M, \tag{1.5.4}$$

where $\mathcal{L}(w, \theta) \coloneqq \sum_{n=1}^N w_n f_n(\theta)$ is the weighted coreset log-likelihood. The coreset is constructed using an important sampling approach, by first computing the *sensitivities* of each data point. These are related to the idea of redundancy introduced earlier, with a larger sensitivity implying a less redundant data point. The sensitivities are given by $\sigma_n \coloneqq \|f_n(\theta)\|_\infty$, using the weighted uniform norm defined above.

In order to deliberately include more points with a lower redundancy in the coreset, the original dataset is then subsampled using an importance sampling approach, by taking M independent draws with probability proportional to σ_n, and then calculating the coreset weights as follows:

$$\sigma \coloneqq \sum_{n=1}^N \sigma_n \quad (M_1, \ldots, M_N) \sim \text{Multi}\left(M, \left(\frac{\sigma_n}{\sigma}\right)_{n=1}^N\right) \quad w_n = \frac{\sigma}{\sigma_n}\frac{M_n}{M}. \tag{1.5.5}$$

In order to calculate these sensitivities in practice, a clustering algorithm such as k-means++ (Arthur and Vassilvitskii, 2006) is used. However, the quality of this initial clustering limits the quality of the coreset approximation, and in practice the approach only improves upon simple uniform sampling by a fixed constant (Campbell and Broderick, 2019). Furthermore, an artificial condition $\Theta = \mathbb{B}_R$ is placed on the state space, where $\mathbb{B}_R \coloneqq \{\theta \in \mathbb{R}^D : \|\theta\|_2 \leq R\}$. This condition is in fact necessary, since the (weighted) uniform norm can diverge in an unbounded parameter space. This precludes many common models which have unbounded parameter spaces.

Another problem, which motivates subsequent approaches, is that the norm lacks a sense of "directionality" as it does not correspond to an inner product. Thus, the algorithm has no way to choose vectors f_n that are aligned to $\mathcal{L}$. This can lead to counter-intuitive scenarios where vectors with a large norm pointed away from $\mathcal{L}$ are chosen, even if there is a vector f_n which is very closely aligned to $\mathcal{L}$. These vectors may be chosen because they have a large weighted uniform norm, and so are included in the coreset to minimize the error in the worst-case setting.

1.5.2 Coresets via Sparse Regression

The first step to developing more efficient and intuitive coreset construction algorithms is to replace the uniform norm by a Hilbert norm. This is a type of norm that corresponds to an inner product, i.e. if $\|\cdot\|$ is a Hilbert norm, then for a given f we can write $\|f\|^2 = \langle f, f\rangle$. The directionality of the Hilbert norm means that points can be intelligently chosen based on residual posterior approximation error (Campbell and Broderick, 2019). Thus, a coreset can be built iteratively; at each step the vector most aligned with the direction that gives the greatest improvement in posterior approximation error is selected, and the corresponding data point is added to the coreset. The first step here is to define $K \in \mathbb{R}^{N\times N}$ as the kernel matrix with $K_{ij} \coloneqq \langle f_i(\theta), f_j(\theta)\rangle$. Then, the fact we are working with a Hilbert norm is utilised to rewrite the objective function as

$$\|\mathcal{L}(w,\theta) - \mathcal{L}(\theta)\|^2 = (w-1)^T K (w-1).$$

The coreset construction problem can then be written as

$$\min_{w\in\mathbb{R}^N} (w-1)^T K (w-1) \qquad \text{s.t. } w \geq 0, \sum_{n=1}^N \sigma_n w_n = \sigma, \tag{1.5.6}$$

where $\sigma_n \coloneqq \|f_n(\theta)\|$ is now defined using a (thus far generic) Hilbert norm, and as before $\sigma \coloneqq \sum_{n=1}^N \sigma_n$. Thus, the problem has been reduced to optimizing a convex function over the polytope defined by $\sum_{n=1}^N \sigma_n w_n = \sigma$, which can be solved using the Frank-Wolfe algorithm (Frank et al., 1956).

The reformulation above holds for any Hilbert norm $\|\cdot\|$, with corresponding inner product $\langle\cdot,\cdot\rangle$. In order to implement this approach in practice, a suitable Hilbert norm needs to be chosen. One choice for this is the weighted L^2 norm on the log-likelihoods, under a weighting distribution $\hat{\pi}$,

$$\|\mathcal{L}(w,\theta) - \mathcal{L}(\theta)\|_{\hat{\pi},2}^2 \coloneqq \mathbb{E}_{\hat{\pi}}\left[(\mathcal{L}(w,\theta) - \mathcal{L}(\theta))^2\right] \tag{1.5.7}$$

$$= \int (\mathcal{L}(w,\theta) - \mathcal{L}(\theta))^2 \hat{\pi}(\theta) d\theta. \tag{1.5.8}$$

This norm induces the inner product

$$\langle f_n(\theta), f_m(\theta)\rangle_{\hat{\pi},2} \coloneqq \mathbb{E}_{\hat{\pi}}\left[f_n(\theta)f_m(\theta)\right] \tag{1.5.9}$$

$$= \int f_n(\theta)f_m(\theta)\hat{\pi}(\theta)d\theta. \tag{1.5.10}$$

Ideally, we want the weighting distribution $\hat{\pi}$ to be chosen equal to π. However, the aim of the overall coreset method is to avoid the significant computational cost associated with accessing the true posterior. Therefore, we instead need to choose a low-cost approximation $\hat{\pi}$. There are several options for this, including the Laplace approximation (Bishop, 2006). This requirement is not uncommon in the literature. However, the quality of the approximation $\hat{\pi}$ fundamentally limits the quality of the coreset approximation, and a key part of subsequent approaches is to remove the need for it.

Campbell and Broderick (2018) note that the construction of Campbell and Broderick (2019) leads to a suboptimally scaling of the coreset log-likelihood in order to maintain feasibility. Instead, $\mathcal{L}(w,\theta)$ should be rescaled to have norm $\|\mathcal{L}(\theta)\|$, and thus they define the normalised vectors

$$\ell_n(\theta) \coloneqq \frac{f_n(\theta)}{\|f_n(\theta)\|}, \quad \ell(\theta) \coloneqq \frac{\mathcal{L}(\theta)}{\|\mathcal{L}(\theta)\|}, \quad \ell(w,\theta) \coloneqq \sum_{n=1}^{N} w_n \ell_n(\theta).$$

The coreset construction problem is therefore equivalent to maximizing the alignment of $\ell(\theta)$ and $\ell(w,\theta)$ over $\ell(w,\theta)$ lying on the unit hypersphere

$$\max_{w\in\mathbb{R}^N} \langle \ell(w,\theta), \ell(\theta)\rangle \qquad \text{s.t. } w \geq 0, \|w\|_0 \leq M, \|\ell(w,\theta)\| = 1. \tag{1.5.11}$$

As before, a suitable Hilbert norm $\|\cdot\|$ needs to be chosen, such as the $\hat{\pi}$-weighted L^2 norm. The coreset construction algorithm is very similar to that of Campbell and Broderick (2019), and shares the drawback that we still need to find a good approximation $\hat{\pi}$ in order to compute the norms and inner products.

1.5.3 Coresets via Sparse Variational Inference

Campbell and Beronov (2019) seek to solve the problems noted in the above approaches by reformulating the coreset construction problem. Rather than relying

on a Hilbert norm, weights are selected to minimize the KL Divergence between the coreset posterior π_w and true posterior π_1:

$$w^\star = \arg\min_{w \in \mathbb{R}^N} D_{KL}(\pi_w || \pi_1) \qquad \text{s.t. } w \geq 0, \|w\|_0 \leq M. \tag{1.5.12}$$

Starting with a coreset of size 1, a greedy algorithm can be used here. At each iteration we select the potential f_n that gives the largest local decrease in KL Divergence around w, and add the corresponding data point to the coreset, before re-optimizing the weights. Choosing this potentially involves the use of Monte Carlo to sample from π_w. This would normally be just as hard as the original problem we are trying to solve. However, the key insight is that if w is built up incrementally, the iterates are sparse, so obtaining samples from π_w has a complexity that scales with M rather than N.

Having selected a potential $f_{n^\star}$ to add, the final step is to use the form of the gradient of D_{KL} to optimize the weights via Stochastic Gradient Descent (SGD). The resulting variational inference algorithm can be shown empirically to outperform the alternatives. Furthermore, it does not require finding a low-cost approximation $\hat{\pi}$, which may be difficult in practice. However, it is prohibitively slow to run, taking several orders of magnitude longer to run than the previous alternatives. In particular, constructing a coreset in this way can take more time than simply performing inference on the full dataset.

1.5.4 Other Bayesian Coreset Construction Methods

The coreset construction methods introduced so far can have limitations in high dimensional settings. In fact, Manousakas et al. (2020) show that in some settings the KL divergences between the optimal coreset posterior and full posterior grows with the dimension. Thus, they seek to address this by introducing the idea of a *pseudocoreset*, for which the approximate posterior takes the form

$$\pi_{u,w}(\theta) = \frac{1}{Z(u,w)} \exp\left(\sum_{m=1}^{M} w_m f\left(u_m, \theta\right)\right) \pi_0(\theta)$$

where $u := (u_m)_{m=1}^M$ are M pseudodata points $u_m \in \mathbb{R}^d$, and $Z(u,w)$ is the corresponding normalization constant. The u_m replace the data, and $f\left(u_m, \theta\right) =$

$\log \pi\left(u_m \mid \theta\right)$ are the new log-likelihood functions. We can then construct a coreset by minimizing the KL divergence over both the pseudodata locations and weights,

$$u^\star, w^\star = \underset{u \in \mathbb{R}^{d \times M}, w \in \mathbb{R}_+^M}{\arg\min} \mathrm{D}_{KL}\left(\pi_{u,w} \| \pi\right).$$

Here, the sparsity constraint is introduced directly through the selection of exactly M pseudo-data points. This approach allows for accurate summarization of data even in the high-dimensional setting, but shares the computational speed issues of the sparse variational inference approach.

One final coreset construction method that we mention here is that of Zhang et al. (2020), which is another approach based on sparse regression. In this case, the optimization problem is solved using the accelerated optimization method Iterative Hard Thresholding (IHT) (Zhang et al., 2019). This method improves upon the performance of the sparse variational inference approach of Campbell and Beronov (2019), whilst having a computational time comparable to Campbell and Broderick (2018). However, it shares the problems of the other sparse regression approaches, namely that it is fundamentally limited by the need to specify a low-cost approximation $\hat{\pi}$.

In Chapter 4, we present a coreset construction algorithm to address several of the drawbacks noted above. The key insight in our work is that we can decouple the problem of selecting a subset of points to constitute the coreset from the problem of choosing their weights. Thus, we can simply select the points to include in the coreset using uniform subsampling. Having selected these points, we then optimize their weights using a novel quasi-Newton method. Construction times are far reduced from the sparse variational inference approach of Campbell and Broderick (2019). Moreover, there is no need for a low-cost approximation $\hat{\pi}$.

1.6 Contributions

In the following chapters, we present three publications relating to the areas of Bayesian statistics introduced above. The papers are presented in their complete form, with individual bibliographies and appendices. In each case, I am the first

author, and have contributed to the theoretical and methodological developments, experiments and paper writing. After each of the papers there is a signed statement of authorship which details my contributions.

1.6.1 Bayesian Nonparametric Network Models

In Chapters 2 and 3 we develop two novel Bayesian nonparametric models for statistical network analysis. In the first publication, we present a model for sparse networks with core-periphery structure. We define what it means for a graph to have this structure in our framework, and introduce a class of networks with such a structure. We also provide theoretical results on the relevant sparsity properties. We then provide methods to simulate from core-periphery networks in our model, and introduce a posterior inference algorithm. We use this to show that our method can detect core-periphery structure in simulated and real networks, and compares favourably against contemporary alternatives when used as a classification algorithm.

This paper has been published in the *Electronic Journal of Statistics*, and can be referenced as follows:

- **Naik, C.**, Caron, F. and Rousseau, J., 2021. Sparse networks with core-periphery structure. *Electronic Journal of Statistics, 15(1), pp.1814-1868. https://doi.org/10.1214/21-EJS1819*

The second publication concerns the development of Bayesian nonparametric models for sparse dynamic networks. Here, we use a dynamic point process model to capture the evolution of sparse networks over time, where edges can appear and disappear. Our model is able to capture the long-term evolution of sociabilities of nodes within the network, and yields sparse graphs. We provide theoretical results on the sparsity properties of the model, and also introduce an approximate posterior inference algorithm. Finally, we apply the model to simulated and real networks, using our posterior inference algorithm to model the changing sociabilities of the nodes.

This paper has been submitted to the *European Conference on Machine Learning and Data Mining (2022)*, and can be referenced as follows:

- **Naik, C.**, Caron, F., Rousseau, J., Teh, Y.W. and Palla, K., 2022. Bayesian nonparametrics for sparse dynamic networks. *arXiv preprint arXiv:1607.016245.*

1.6.2 Bayesian Coresets

In Chapter 4, we study the problem of data subsampling as a preprocessing step to Bayesian inference. This publication introduces a novel Bayesian coreset construction algorithm. Our algorithm simplifies previous approaches, and removes a significant amount of user input, by first forming a coreset by simple uniformly sampling, and then optimizing the weights on those points using a novel quasi-Newton method. We provide theoretical results to show that our method finds a high-quality coreset with high probability. We also show empirically that our method provides a significant reduction in construction time against black-box methods. Moreover, it compares favourably against realistic alternatives, with far less storage cost required.

This paper has been submitted to the *International Conference on Machine Learning (2022)*, and can be referenced as follows:

- **Naik, C.**, Rousseau, J. and Campbell, T., 2022. Fast Bayesian Coresets via Subsampling and Quasi-Newton Refinement. *arXiv preprint arXiv:2203.09675.*

Aside note

During my DPhil I also worked on assessing how geographical diversity affects collaborative research. This paper is not included in this thesis, but is currently in preprint form and can be referenced as follows:

- **Naik, C.**, Sugimoto, C. R., Larivière, V., Leng, C. and Guo, W., 2022. Impact of Geographic Diversity on Citation of Collaborative Research. *arXiv preprint arXiv:2204.11713.*

2

Sparse Networks with Core-Periphery Structure

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

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

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

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

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 $w_{ik} > 0$, $k = 2, \ldots, K$. 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:

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

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

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

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

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

where F is a probability distribution on $\mathbb{R}_+^{K-1}$ and the scores $\beta_{i1} \in \{0,1\}$ and $(\beta_{i2},\ldots,\beta_{iK}) \in \mathbb{R}_+^{K-1} > 0,\ i \geq 1$ 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:

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

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].

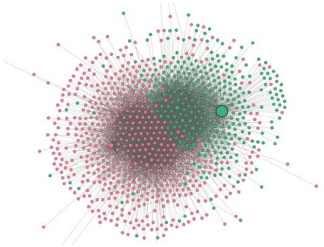
Note crucially that, although the formulation (2.10) resembles the formulation of the class of compound CRMs introduced by [17], in our construction, the scores β_{i1} are not identically distributed, and depend on the base parameter w_{i0}. This key difference enables the model to have different sparsity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

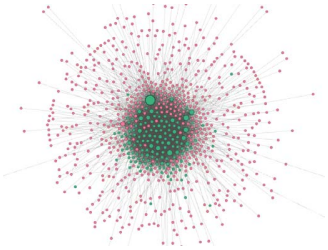
$$F(d\beta_2,\ldots,d\beta_K) = \prod_{k=2}^{K} \beta_k^{a_k-1} e^{-b_k\beta_k} \frac{b_k^{a_k}}{\Gamma(a_k)} d\beta_k, \tag{2.13}$$

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

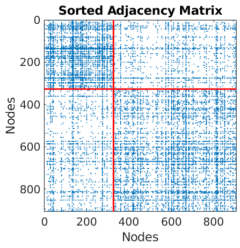
We simulate a network from the above model with parameters $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = 0.5, a = 0.2$. This corresponds to the case where we have core-periphery but not community structure. In either case, we refer to our model as the Sparse Core Periphery (SparseCP) model. For comparison,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



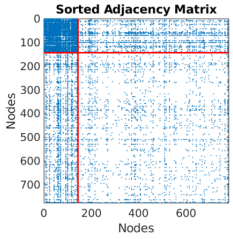
(a) SNetOC graph with 2 communities



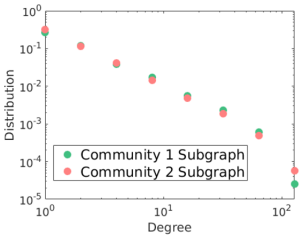
(b) SparseCP graph with core-periphery stucture



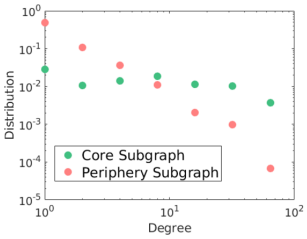
(c) SNetOC adjacency matrix



(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

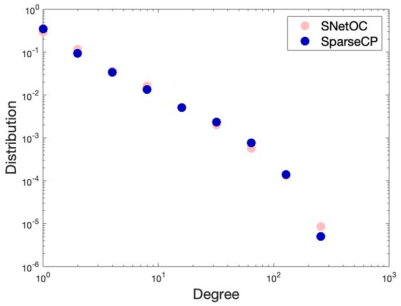


FIG 2. *Overall degree distribution of the SNetOC and SparseCP graphs.*

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.
- $\mathbf{w_{i2}}, \ldots, \mathbf{w_{iK}}$ — When $K = 2$, we can interpret w_{i2} as an overall sociability parameter. Otherwise $w_{i2}, \ldots, w_{iK}$ 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.

- $\mathbf{a_2}, \ldots, \mathbf{a_K}, \mathbf{b_2}, \ldots, \mathbf{b_K}$ — 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*

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

If

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

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

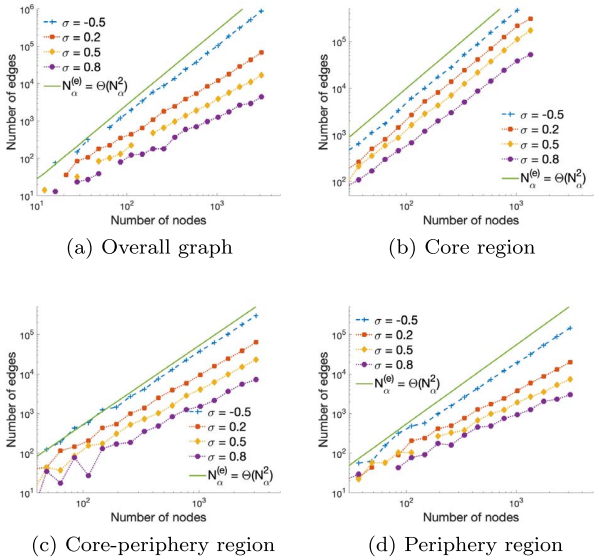
$$\frac{N_{\alpha,c}}{N_\alpha} = O\left(\alpha^{-\sigma}\right) \qquad \text{almost surely as } \alpha \to \infty \tag{3.5}$$

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

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

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

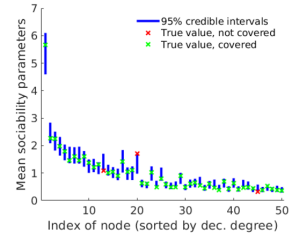
We start by considering simulated data. This is firstly in order to study the convergence of our MCMC algorithm. Secondly, it allows us to compare our model to that of [41] in a setting where have complete information about the ground truth.

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.

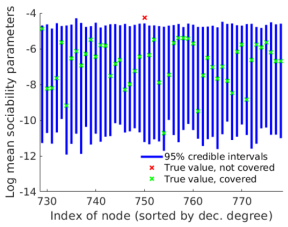
We fit our model to the simulated network, placing vague $\Gamma(0.01, 0.01)$ priors on the unknown parameters α, $1-\sigma$, τ, a_k and b_k. We initialize the parameters with an initialization run (as described in Section C.4 of Appendix C) of 10 000 steps. We then run 3 parallel MCMC chains of 500 000, 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.

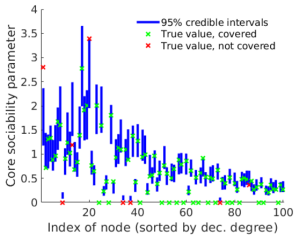
Moreover, we see that we are able to very accurately recover the classification into core and periphery. In our generated graph, 137 out of 144 nodes are classified correctly into the core, with 631 out of 634 are classified correctly into the periphery. Importantly, the core is not simply comprised 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



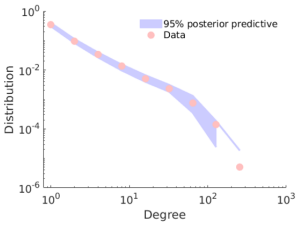
(a) Mean sociability credible intervals for high degree nodes



(b) Mean sociability credible intervals for low degree nodes



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



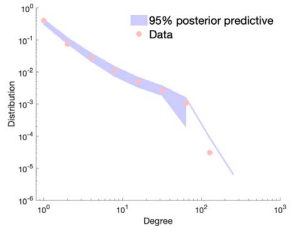
(d) Posterior predictive degree distribution

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.*

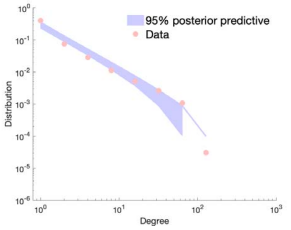
with $w_{i1} = 0$, can have a high probability of connection with another node j if they both have large overall sociabilities w_{i2}, w_{j2}. Thus, nodes not in the core can still have a high degree. Conversely, nodes in the core but with low overall sociabilities may have low degrees. In Figure 4(c) 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.

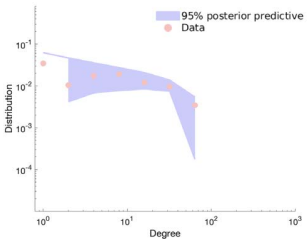
Comparison with the SNetOC model of [41]. Our model is an extension of that of [41], to allow the modelling of core-periphery structure as well as community structure. The question arises as to whether their SNetOC model could



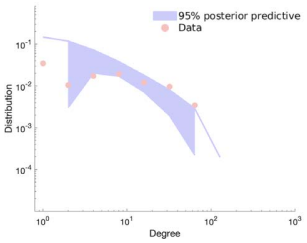
(a) SparseCP posterior predictive degree distribution



(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

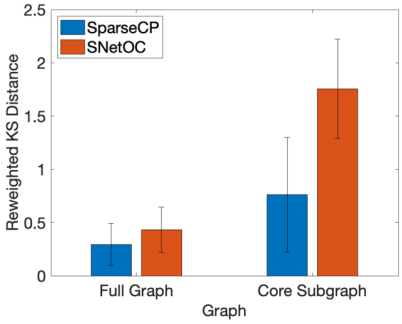


FIG 6. *Reweighted Kolmogorov Smirnov distance for assessing the degree distribution for the network generated from the SparseCP model with parameters* $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = 1, a = 0.2$.

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.

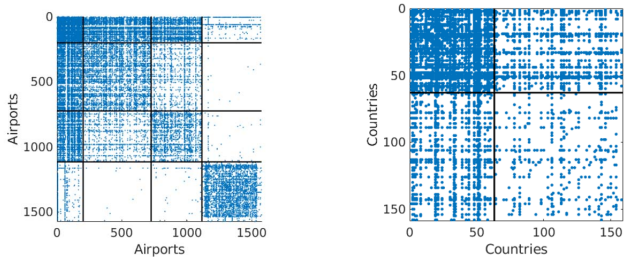
TABLE 1
Real World Networks and Summary Statistics.

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.*

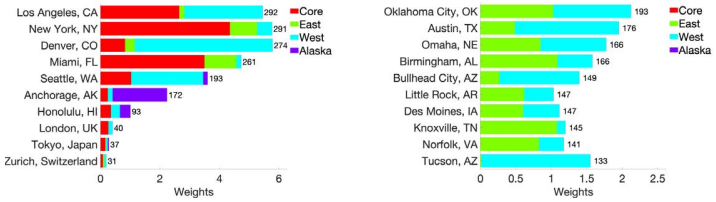
If we investigate the classification into core and periphery in more detail, we see several interesting features. Firstly, when looking at the airports placed in the core, we can compare these to the list of hub airports for various airline companies[4]. We see that 38 out of 47 of these airline hub 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.*

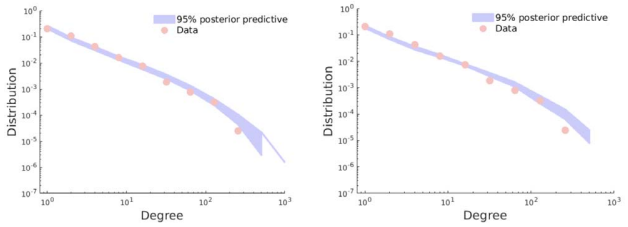
ter entering the gamma distribution F corresponding to the Alaskan community (the 3rd community identified by our model) concentrates on values very close to 0 (see Figure 33 in Appendix D). This increases the posterior variance of the sociabilities w_{i4}, and leads to some graphs with very high degree 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

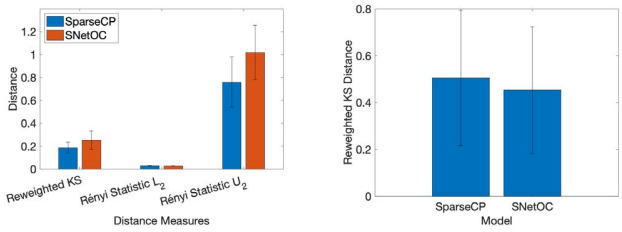
$$L_2 = \sup_{x \geq x_{min}} \frac{P(x)}{S(x)}, \quad U_2 = \sup_{x \leq x_{max}} \frac{1 - P(x)}{1 - S(x)} \tag{4.3}$$



(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

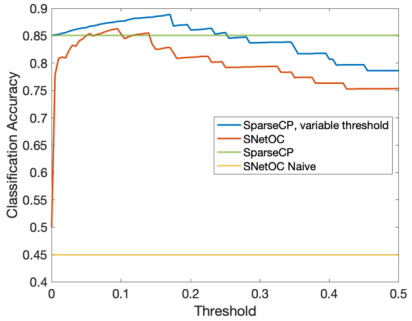


FIG 11. *Classification accuracy of the airline hubs in the US Airport network for the SparseCP and SNetOC models with varying threshold.*

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

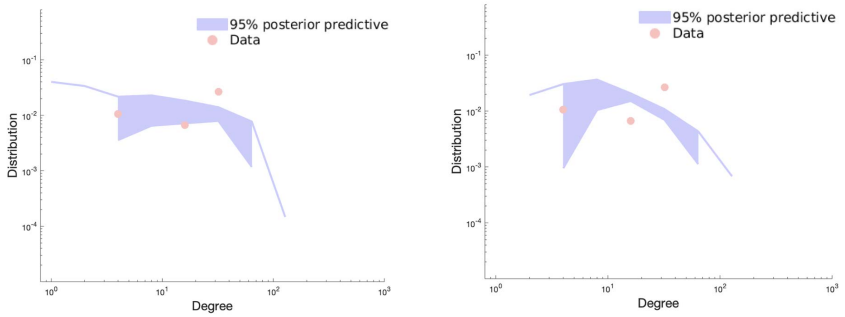
Figure 11 compares the classification accuracy from the SparseCP model (in green) to that obtained from the SNetOC model with different thresholds (in red). We see that with the right choice of threshold, the SNetOC model can slightly outperform ours, but for other thresholds it performs far worse. We 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

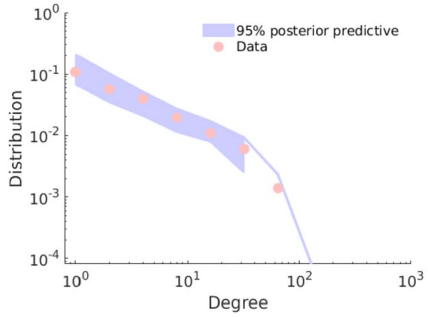


FIG 13. *Posterior predictive degree distribution for the World Trade Network, using the SparseCP model. When the lower posterior predictive interval bound is 0, we plot only the upper interval bound.*

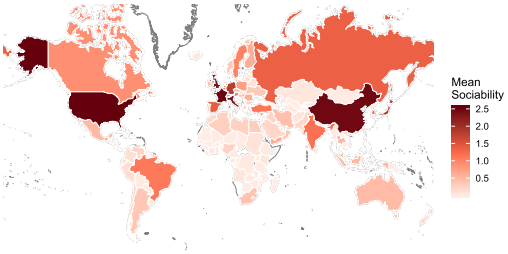
parameter in Figure 14(a), and by the value of the core sociability parameter in Figure 14(b). We see that the core consists largely of the large, developed countries that we expect, as well as some smaller European countries. This fits with other results that have been obtained in the literature [12]. 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

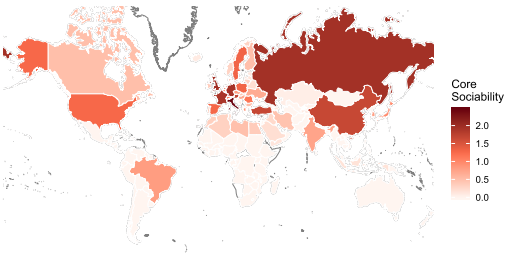
Finally, we want to compare our method against other standard methods of core-periphery detection. In order to do this we generate simulated core-periphery networks, and run each algorithm on them. We can then calculate the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve [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.

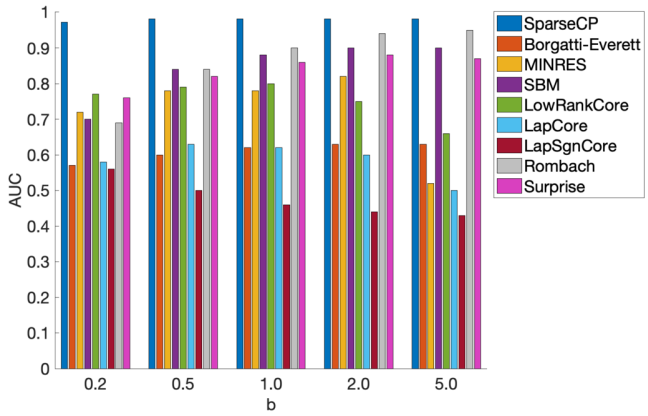


FIG 15. *AUC of different classification methods for the graph generated from the SparseCP model with parameters* $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, a = 0.2$*, and differing values of* b*.*

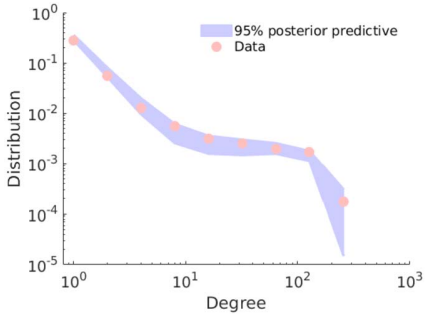


FIG 16. *Posterior predictive degree distribution using the SparseCP model for the graph generated from the SparseCP model with parameters* $K = 2, \alpha = 200, \sigma = 0.2, \tau = 1, b = 5, a = 0.2$*.*

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 clas-

sify 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.

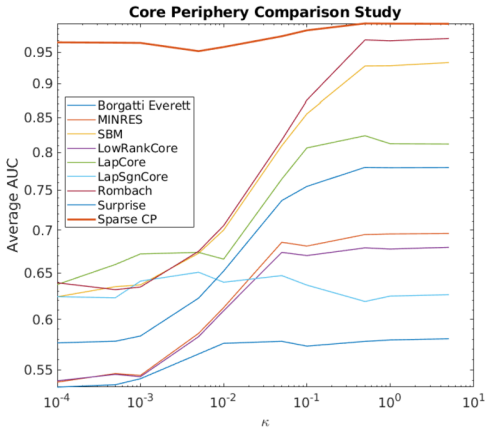


FIG 17. *Average classification accuracy (as measured by the AUC) for a range of standard methods for core-periphery detection, along with classification accuracy using the SparseCP model. The accuracies are calculated for different values of the parameter κ, which controls the extent to which core and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

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

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

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

Let

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

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

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

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 $\int_{(0,\infty)^K}\rho(dw_1, dw_2, \ldots, dw_K) < \infty$ (by assumption (2.9)). By the the law of large numbers, $Q_{\alpha,c} \sim \alpha\int_{(0,\infty)^K}\rho(dw_1, dw_2, \ldots, dw_K)$. 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 $\int_{\{0\}\times(0,\infty)^K}\rho(dw_1, dw_2, \ldots, dw_K) < \infty$, 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 $(w_{i2}, \ldots, w_{iK})_{i \geq 1, w_{i1}=0}$ are the points of a Poisson point process with mean $\rho_p(dw_2, \ldots, dw_K) = \int_{w_1 \in [0,\infty)} \rho_p(dw_1, dw_2, \ldots, dw_K)$. This is therefore the same model as that of [41]. We therefore have, using Proposition 4 in [41]

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

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

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

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

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

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

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

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

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

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

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

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

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.

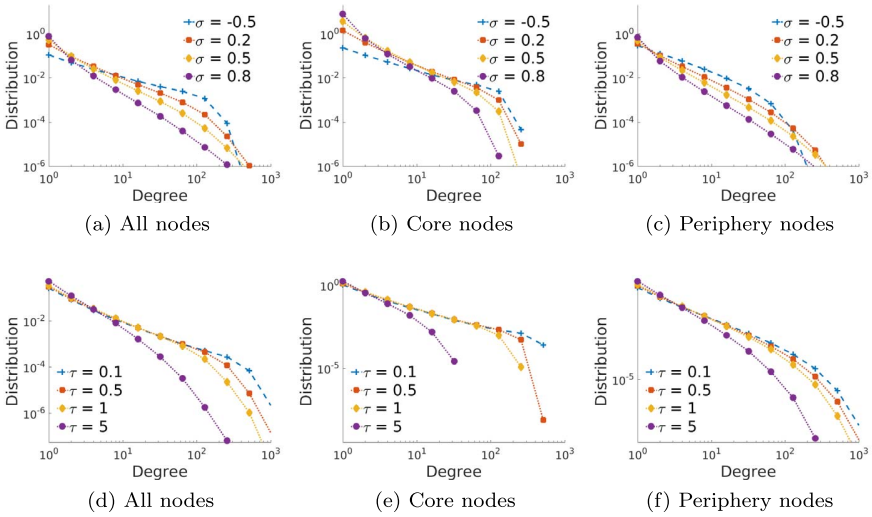
Figure 19 shows the results for varying a and b. When increasing a we see that the shape of the degree distribution for the core does not change, although the number of nodes in the core increases. Overall and in the periphery we see little difference in the degree distribution apart from nodes with 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

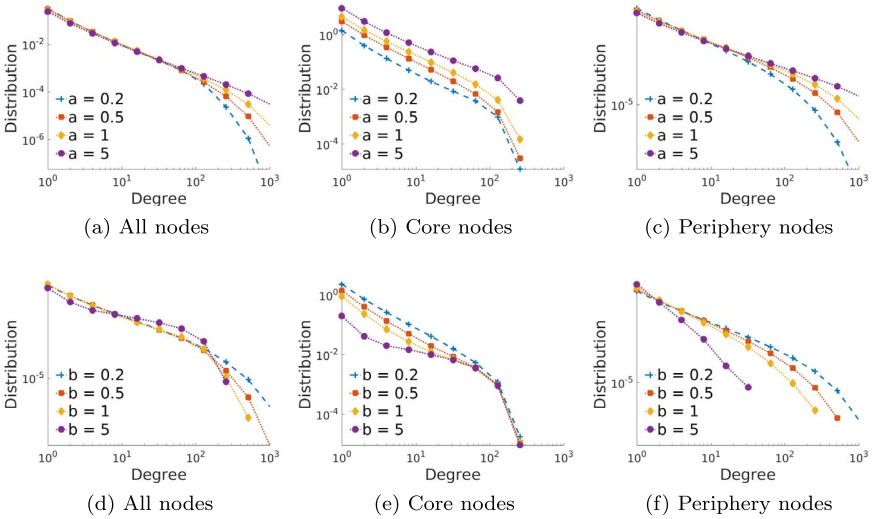
We are finally interested in how the proportion of nodes in the core is affected by changing a and b. From Figure 22 we see that, as expected, the asymptotic rates are not affected by changing these hyperparameters. However, for fixed size graphs, increasing a decreases the relative size of the core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



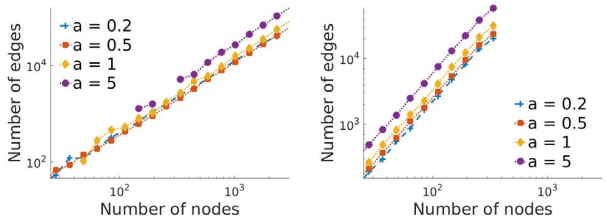
(a) All nodes (b) Core nodes (c) Periphery nodes

(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

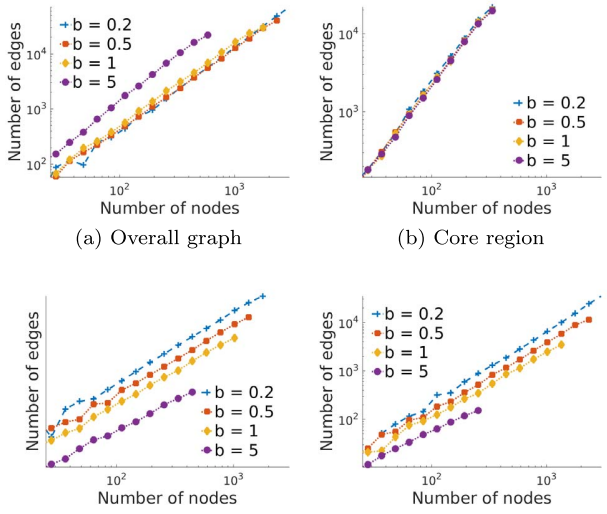
We give here the details of our MCMC algorithm, as detailed in Algorithm 1, in particular how it differs from the Algorithm of [41].



(a) Overall graph (b) Core region

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

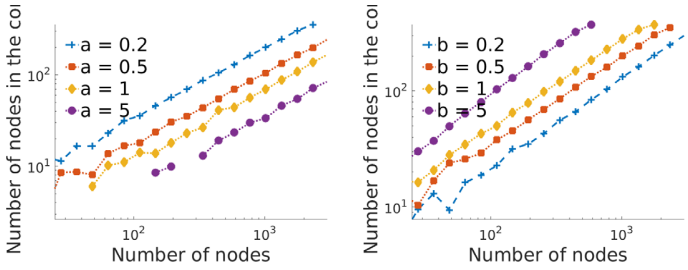
FIG 20. *Relationship between nodes and edges for the* (*a*) *overall graph,* (*b*) *core,* (*c*) *core-periphery region and* (*d*) *periphery of the graph generated from the SparseCP model with varying a.*



(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 $(z_{ij})_{1\le i,j,\le N_\alpha}$ 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

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

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 (ϕ,α) 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]:

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

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

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 $(w_{i0}, \beta_{i1},\ldots,\beta_{iK})_{i=1,\ldots,N_\alpha}$ all at once, with the potential energy function U defined as

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

$U(w_0,\beta)$ 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 $(\beta_{i1})_{i=1,\ldots,N_\alpha}$ cannot be used. We propose an alternative solution, using a Gibbs step to update β_1 separately from the others as follows

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

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 $(w_0,\beta_2,\ldots,\beta_K)$, 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 $(\beta_{i1})_{i=1,\ldots,N_\alpha}$ 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

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

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 $(w_{*1}, \ldots, w_{*K}) = X_\epsilon + X^\epsilon$, with $X_\epsilon = \sum_{i|w_{i0}<\epsilon} w_{i0}(\beta_{i1}, \ldots, \beta_{iK})$ and $X^\epsilon = \sum_{i|w_{i0}>\epsilon} w_{i0}(\beta_{i1}, \ldots, \beta_{iK})$. Then, a realization of X^ϵ can be simulated exactly from a Poisson process with mean measure

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

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

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

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

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$(\lambda_1, \ldots, \lambda_K)$ distribution is

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

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

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

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 $I = \{i_1, \ldots, i_N\},\ i_1 < \ldots < i_N$, 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 $(\widetilde{n}_{ij1}, \ldots, \widetilde{n}_{ijK})$ as normal, using the conditional distribution given z, w.
4. For $(i,j) \in A$, propose an update for $(\widetilde{n}_{ij1}, \ldots, \widetilde{n}_{ijK})$ from the mixture distribution:

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

   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 $\widetilde{n}_{ij2}, \ldots, \widetilde{n}_{ijK}$ from a truncated Poisson distribution. With probability $1 - p_{lat}$, $\widetilde{n}_{ij1}, \ldots, \widetilde{n}_{ijK}$ 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 $\widetilde{n} = (\widetilde{n})_{1 \leq i,j \leq N_\alpha, k=1,\ldots,K}$, 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 $w_k,\ k = 2, \ldots, K$ 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

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

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

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

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

where $\psi(t_1,\ldots,t_K)$ 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

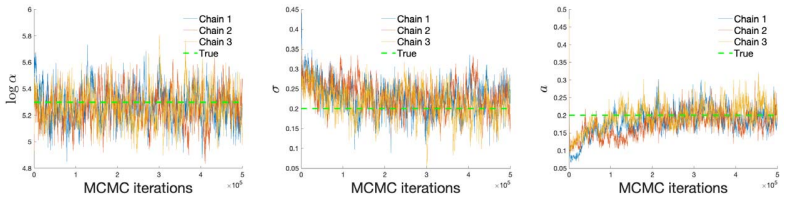
D.1.1. Core-periphery structure only

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.

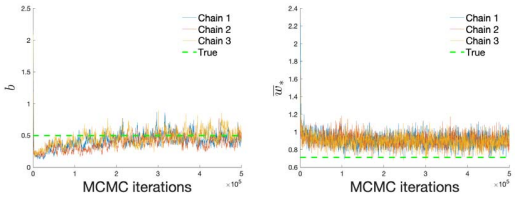
The green lines and stars respectively correspond to the values of the model parameters used to generate the graphs, and we see that the posterior converges around the true values.

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.

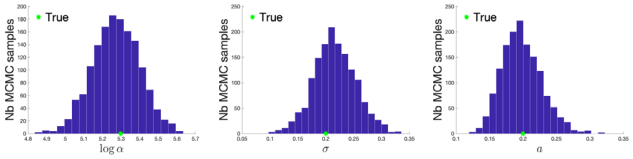


(a) $\log \alpha$ trace (b) σ trace (c) a trace

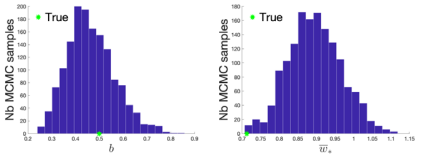


(d) b trace (e) $\bar{w}_*$ trace

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

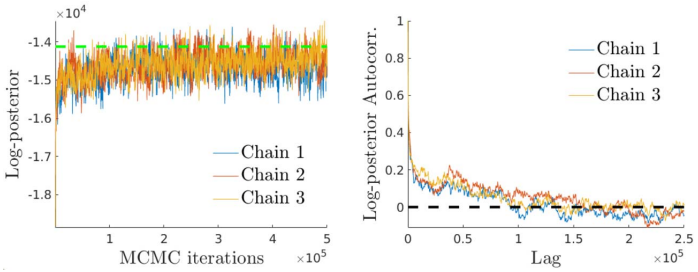


(d) b histogram (e) $\bar{w}_*$ histogram

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.*

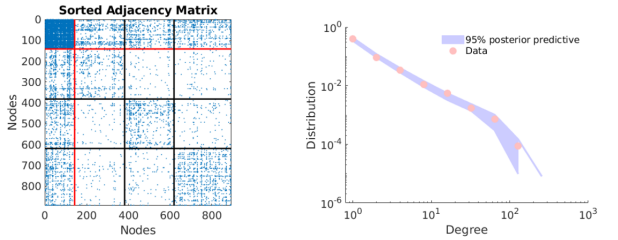
D.1.2. Core-periphery and community structure

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

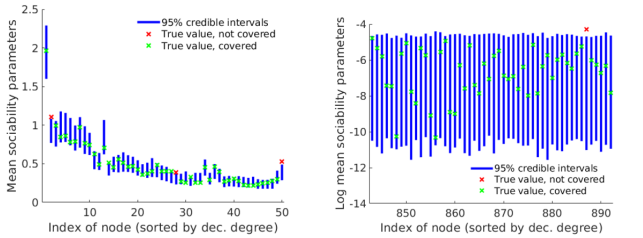


(a) Log-posterior (b) Autocorrelation

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 $w_1, \ldots, w_4$, and see that we are able to recover both the core and community sociabilities.

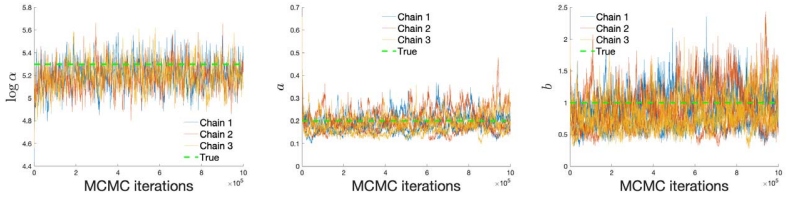
As before, to test convergence we calculate the approximate log-posterior probability density function (up to a normalizing constant). We see from Figure 30 that the chain has converged to the true value. This is confirmed by the Gelman-Rubin diagnostic, which in this case comes out as $\hat{R} = 1.03$. 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.

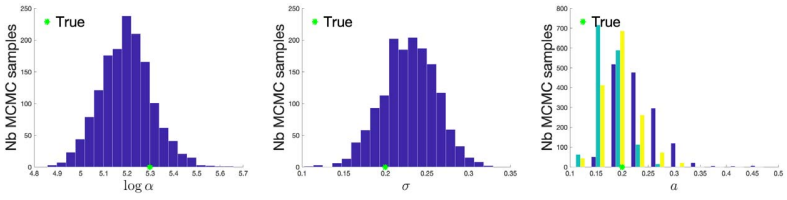
1. σ is (correctly) estimated to be negative, and the size of the core goes to zero. This is the behaviour we see in Figure 31(b) and is the behaviour we would expect.
2. σ is (incorrectly) estimated to be positive, and the whole graph is estimated to be in the core. Although this may not seem intuitive, 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

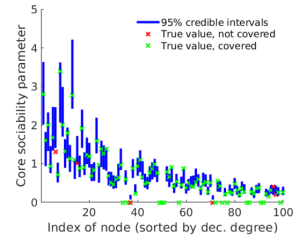
FIG 27. *MCMC trace plots parameters* (*a*) $\log \alpha$, (*b*) σ, (*c*) a, (*d*) b *and* (*e*) $\bar{w}_*$. *The graph is generated from the SparseCP model with* $K = 4, \alpha = 200, \sigma = 0.2, \tau = 1, b_k = b = \frac{4}{K}, a_k = a = 0.2$. *The green lines correspond to the values of the parameters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



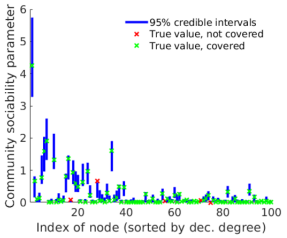
(a) $\log \alpha$ histogram (b) σ histogram (c) a histogram

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

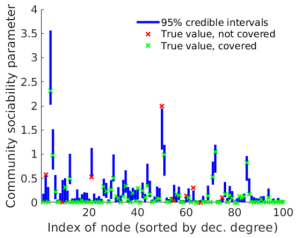
FIG 28. *MCMC histograms for parameters* (*a*) $\log \alpha$, (*b*) σ, (*c*) a, (*d*) b *and* (*e*) $\bar{w}_*$. *The graph is generated from the SparseCP model with* $K = 4, \alpha = 200, \sigma = 0.2, \tau = 1, b_k = b = \frac{4}{K}, a_k = a = 0.2$. *The green stars correspond to the values of the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



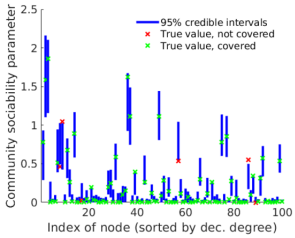
(a) Core sociability credible intervals for high degree nodes



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

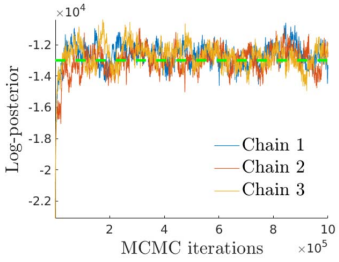


(c) Community 2 sociability credible intervals for high degree nodes

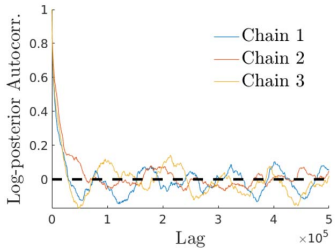


(d) Community 3 sociability credible intervals for high degree nodes

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.*

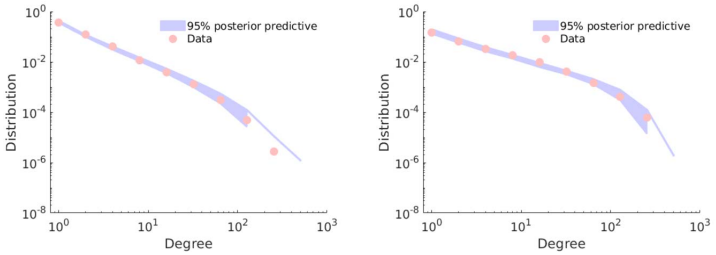


(a) Log-posterior



(b) Autocorrelation

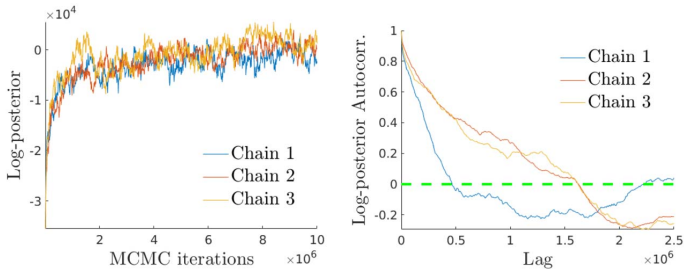
FIG 30. *Trace plot and autocorrelation of the log-posterior for a graph generated from the SparseCP model with* $K = 4, \alpha = 200, \sigma = 0.2, \tau = 1, b_k = b = \frac{4}{K}, a_k = a = 0.2$. *The green line in* (*a*) *corresponds to the value of the approximate log-posterior (up to a constant)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

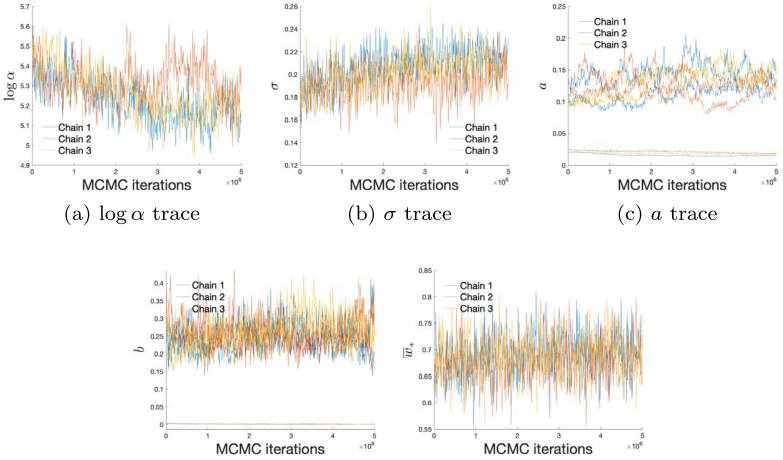
FIG 32. *Trace plot (left) and autocorrelation plot (right) of the approximated log-posterior probability density (up to a constant) of the US Airport network.*

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.

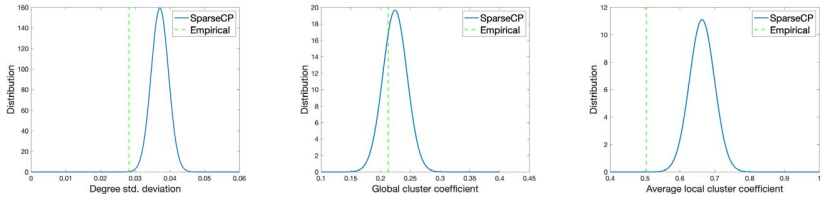
These results indicate that the convergence of our sampler is slow. As in



(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.*

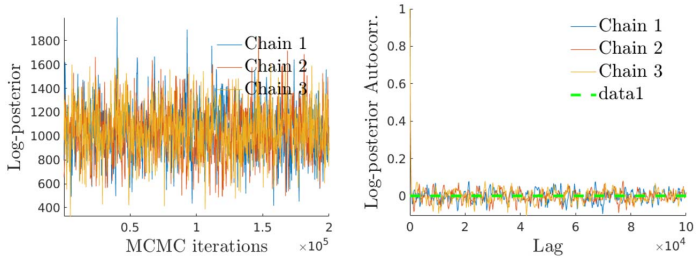


(a) Posterior predictive degree standard deviation (b) Posterior predictive global cluster coefficient (c) Posterior predictive average local cluster coefficient

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.*

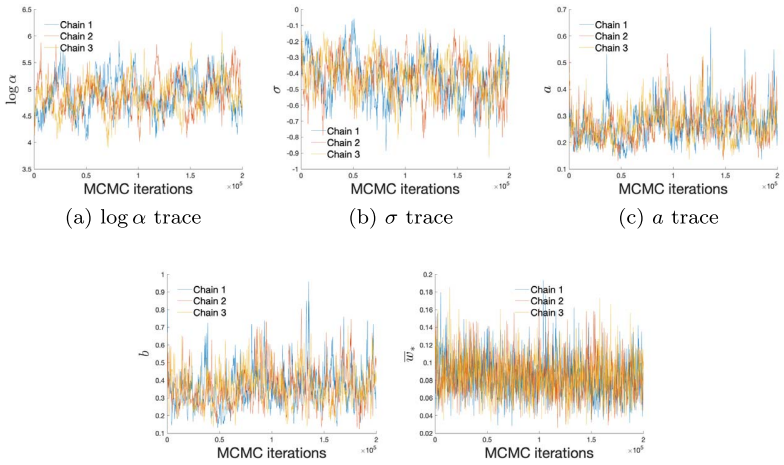
[41], even running a very long chain did not lead to an improvement. This long chain took roughly 60 hours to run, translating to a time cost of around 0.02 seconds per iteration. However, it is at least encouraging to see that the different chains are exploring parts of the parameter space with similar 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

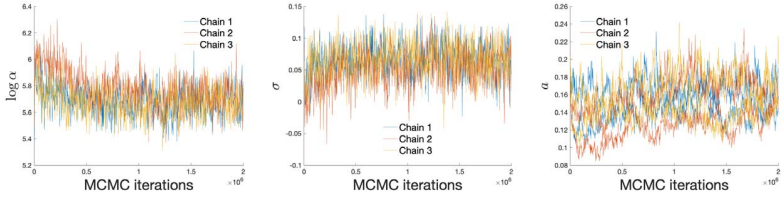
(d) b trace (e) $\bar{w}_*$ trace

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.*

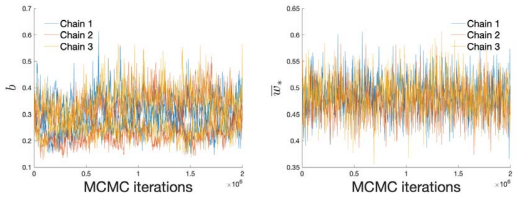
D.2.2. World trade network

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

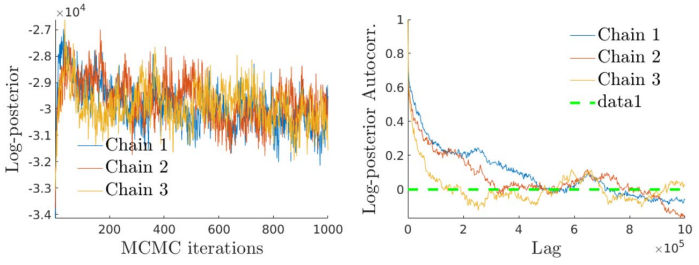
In Section 4.3.1, we note a problem occurring due to the sociability parameter for the Alaskan community. Here, we repeat our analysis having taken out all the Alaskan airports, as well as any airport only connected to Alaskan airports. The relevant plots comprise Figures 37-41.

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.

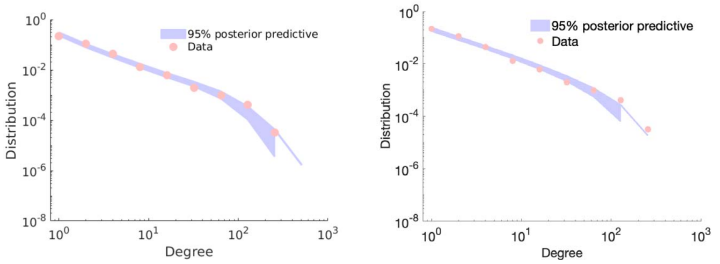
In Figure 39 we see the posterior predictive degree plot on the left, compared to the corresponding plot for the model of [41] with three communities on the right. We see that we are no longer overestimating the high degree nodes. We also see that the SNetOC model has estimated $\sigma < 0$ in this case,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

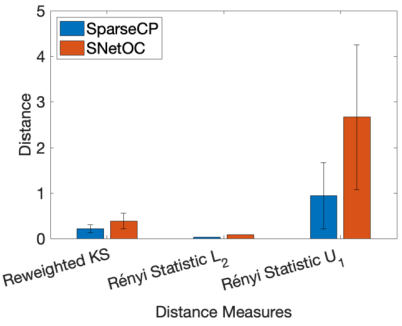


FIG 40. *Reweighted Kolmogorov Smirnov distance and Rényi statistics for assessing the degree distribution for the US Airport network without Alaskan airports.*

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

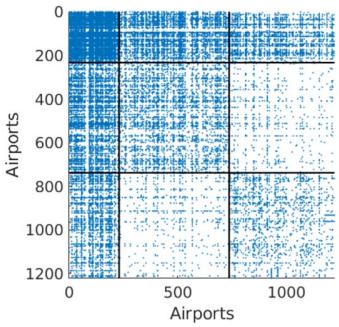


FIG 41. *Adjacency matrix for the US Airport network without Alaskan airports, ordered into core and periphery, and then by highest community weight, as estimated using the SparseCP model.*

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

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Sparse networks with core-periphery structure
Publication Status	☒Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Published in the Electronic Journal of Statistics, 2021.

Student Confirmation

Student Name:	Cian Naik		
Contribution to the Paper	I am the first author of this paper. I have contributed to designing and implementing the inference scheme, running the experiments, developing the theoretical results and writing the paper.		
Signature	Cian Naik	Date	25/03/22

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor François Caron		
Supervisor comments I certify that Cian made a substantial contribution to the publication, as described above.		
Signature	Date	28/03/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

3

Bayesian Nonparametrics for Sparse Dynamic Networks

Bayesian Nonparametrics for Sparse Dynamic Networks

Cian Naik[1], François Caron[1], Judith Rousseau[1], Yee Whye Teh[1,3], and Konstantina Palla[2]

[1] Department of Statistics, University of Oxford, Oxford, United Kingdom
[2] Microsoft Research, Cambridge, United Kingdom
[3] Google Deepmind, London, United Kingdom

Abstract. In this paper we propose a Bayesian nonparametric approach to modelling sparse time-varying networks. A positive parameter is associated to each node of a network, which models the sociability of that node. Sociabilities are assumed to evolve over time, and are modelled via a dynamic point process model. The model is able to capture long term evolution of the sociabilities. Moreover, it yields sparse graphs, where the number of edges grows subquadratically with the number of nodes. The evolution of the sociabilities is described by a tractable time-varying generalised gamma process. We provide some theoretical insights into the model and apply it to three datasets: a simulated network, a network of hyperlinks between communities on Reddit, and a network of co-occurences of words in Reuters news articles after the September 11^{th} attacks.

Keywords: Bayesian nonparametrics · Poisson random measures · networks · random graphs · sparsity · point processes

1 Introduction

This article is concerned with the analysis of dynamic networks, where one observes the evolution of links among a set of objects over time. As an example, links may represent social interactions between individuals over time or the co-occurrence of words in a newspaper over time. Probabilistic approaches treat the dynamic networks of interest as random graphs, where the vertices (nodes) and edges correspond to objects and links respectively. In the graph setting, sparsity is defined in terms of the rate in which the numbers of edges grows as the number of nodes increases. In a *sparse* graph the number of edges grows sub-quadratically in the number of nodes. Hence, in a large graphs, two nodes chosen at random are very unlikely to be linked.

While sparsity is a property found in many real-world network datasets [33], most of the popular Bayesian models used in network analysis account for dense graphs, i.e. where the number of edges grows quadratically in the number of nodes, see [35] for a review. A recent Bayesian nonparametric approach, proposed by [6] and later developed in a number of articles [39, 18, 3, 38, 31] represents the

graph as an infinite point process on $\mathbb{R}_+^2$, giving rise to a class of sparse random graphs. This class of sparse models is projective and admits a representation theorem due to [24].

In this paper, we are interested in the dynamic domain and aim to probabilistically model the evolution of sparse graphs over time, where edges may appear and disappear, and the node popularity may change over time. We build on the sparse graph model of [6] and extend it to deal with time series of network data. We describe a fully generative and projective approach for the construction of sparse dynamic graphs. It is challenging to perform exact inference using the framework we introduce, and thus we consider an approximate inference method, using a finite-dimensional approximation introduced by [28].

The rest of the article is structured as follows. In Section 2 we give some background on the sparse network model of [6]. Section 3 describes the novel statistical dynamic network model we introduce in detail. In Section 4, we describe the sparsity properties of the proposed model. The approximate inference method, based on a truncation of the infinite-dimensional model is described in Section 5. In Section 6 we present illustrations of our approach to three different dynamic networks with thousands of nodes and edges.

2 Background: model of Caron and Fox for sparse static networks

We recall in this section the model of [6] for sparse multigraphs. Let $\alpha > 0$ be a positive real tuning the size of the network. A finite multigraph of size $\alpha > 0$ is represented by a point process on $[0, \alpha]^2$

$$N = \sum_{i,j} n_{ij} \delta_{(\theta_i, \theta_j)}$$

where $n_{ij} = n_{ji} \in \{0, 1, 2, \ldots\}$, $i \leq j$, represents the number of interactions between individuals i and j, and the $\theta_i \in [0, \alpha]$ can be interpreted as node labels. Those node labels are introduced for the model's construction, but are not observed nor inferred. Each node i is assigned a sociability parameter $w_i > 0$. Let $W = \sum_i w_i \delta_{\theta_i}$ be the corresponding random measure on $[0, \alpha]$. We assume that W is a generalised gamma completely random measure [26, 21, 4, 29]. That is, $\{(w_i, \theta_i)_{i \geq 1}\}$ are the points of a Poisson point process with mean measure $\nu(w)dw\mathbb{1}_{\theta \leq \alpha}d\theta$ where $\mathbb{1}_A = 1$ if the statement A is true and 0 otherwise, and ν is a Lévy intensity on $(0, \infty)$ defined as

$$\nu(w) = \frac{1}{\Gamma(1-\sigma)} w^{-1-\sigma} e^{-\tau w} \tag{1}$$

with hyperparameters $\sigma < 1$ and $\tau > 0$. We write simply $W \sim \mathrm{GG}(\alpha, \sigma, \tau)$.

To each pair of nodes i, j, we assign a number of latent interactions n_{ij}, where

$$n_{ij} \mid w_i, w_j \sim \begin{cases} \text{Poisson}(2w_i w_j) \; i < j, & n_{ji} = n_{ij} \\ \text{Poisson}(w_i w_j) \quad i = j \end{cases} \tag{2}$$

Finally, two nodes are said to be connected if they have at least one interaction; let $z_{ij} = \mathbb{1}_{n_{ij}>0}$ be the binary variable indicating if two nodes are connected. When $\sigma > 0$, such model yields sparse graphs with power-law degree distributions [6, 7].

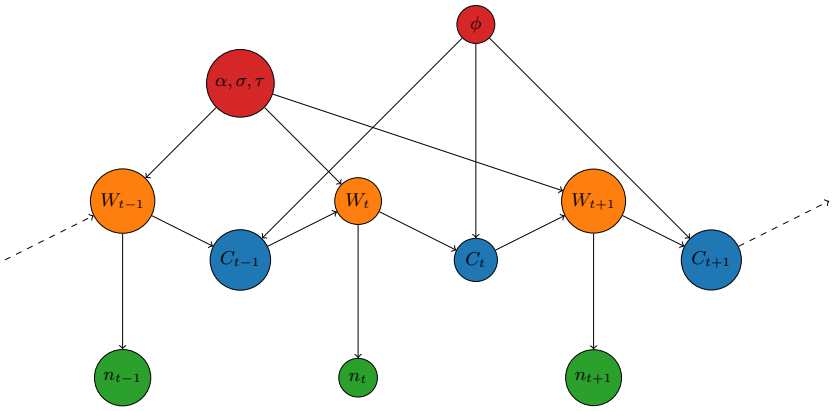


Fig. 1: Graphical representation of the model. The counts n_t are derived from the sociabilities W_t, whose time-evolution depends on the counts C_t and hyperparameters α, σ, τ and ϕ.

3 Dynamic statistical network model

In order to study dynamically evolving networks, we assume that at each time $t = 1, 2, \ldots, T$, we observe a set of interactions between a number of nodes. This set of interaction is represented by a point process N_t over $[0, \alpha]^2$ as in Equation (3), where α tunes the size of the graphs.

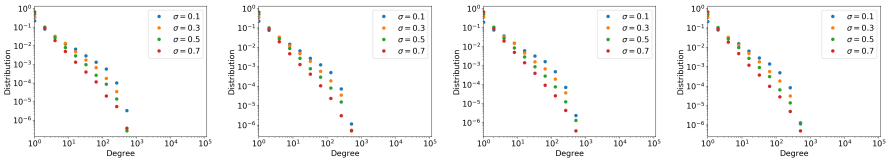
$$N_t = \sum_{i,j} n_{tij} \delta_{(\theta_i, \theta_j)}. \tag{3}$$

Here, n_{tij} is the number of interactions between i and j at time t, and the θ_i are unique node labels.

The dynamic point process N_t is obtained as follows. We assume that each node i at time t has a *sociability* parameter $w_{ti} \in \mathbb{R}_+$, that can be thought of as a measure of the node's willingness to interact with other nodes at time t. We consider the associated collection of random measures on $\mathbb{R}_+$, for $t = 1, \ldots, T$

$$W_t = \sum_i w_{ti} \delta_{\theta_i}, \quad t = 1, \ldots, T.$$

We first describe in Section 3.1 the model for the latent interactions. Then we describe in Section 3.2 the model for the time-varying sociability parameters $(W_t)_{t \geq 1}$. The overall probabilistic model is summarised in Figure 1.



(a) $t = 1$ (b) $t = 2$ (c) $t = 3$ (d) $t = 4$

Fig. 2: Degree distributions over time, for a network simulated from the GG model with $T = 4$, $\alpha = 200$, $\tau = 1$, $\phi = 1$ and varying values of σ.

3.1 Dynamic network model based on observed interactions

In the dynamic setting, what we observe in practice is often counts of interactions between nodes, e.g. hyperlinks, emails or co-occurrences, rather than a binary indicator of whether there is a connection between them. So for each pair of nodes $i \leq j$, we let $(n_{tij})_{t=1,2,\ldots T, j \geq i}$ be the interaction count between them at time t. We assume that n_{tij} can be modelled as

$$n_{tij} \mid w_{ti}, w_{tj} \sim \begin{cases} \text{Poisson}(2w_{ti}w_{tj}) \; i < j, & n_{tji} = n_{tij} \\ \text{Poisson}(w_{ti}w_{tj}) \quad i = j \end{cases} \tag{4}$$

This model can be easily adapted to graphs with directed edges, by modifying the Equation (4) to

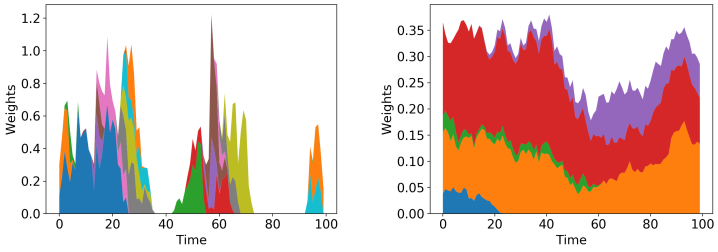
$$n_{tij} \sim Poisson(w_{ti}w_{tj})$$

for all $i \neq j$, where n_{tij} now represents the number of interactions from i to j. The resulting inference algorithm essentially remains the same, and from now on we assume we are in the undirected edge setting. As in the static case, we can reconstruct the binary graph by letting $z_{tij} = \mathbb{1}_{(n_{tij}>0)}$ be the binary variable indicating if nodes i and j are connected at time t, i.e. two nodes are connected at time t if and only if $n_{tij} > 0$. To avoid ambiguity, we say that the number of edges in the graph at time t is $\sum_{i>j} z_{tij}$, rather than counting the number of interactions between pairs of nodes. Marginalizing out the interaction counts n_{tij}, we have for $i \neq j$:

$$\Pr(z_{tij} = 1 \mid (w_{t-k,i}, w_{t-k,j})_{k=0,\ldots,t-1}) = 1 - e^{-2w_{ti}w_{tj}}. \tag{5}$$

3.2 A dependent generalised gamma process for the sociability parameters

We consider here that the sequence of random measures $(W_t)_{t=1,2,\ldots}$ follows a Markov model, such that W_t is marginally distributed as $\text{GG}(\alpha, \sigma, \tau)$. To this aim, we build on the generic construction of [36]. A similar model has been derived by [8] for dependent gamma processes (corresponding to the case $\sigma = 0$ here). As in [6], we use the generalised gamma process here because of the



(a) $\phi = 20$ (b) $\phi = 2000$

Fig. 3: Evolution of weights over time, for a network simulated from the GG model with $T = 100$, $\alpha = 1$, $\sigma = 0.01$, $\tau = 1$ and (a) $\phi = 20$ (b) $\phi = 2000$.

flexibility the sparsity parameter σ gives us. In particular, this setup allows us to capture power-law degree distributions, unlike with the gamma process.

For a sequence of additional latent variables $(C_t)_{t=1,2,\ldots}$, we consider a Markov chain $W_t \to C_t \to W_{t+1}$ starting with $W_1 \sim \mathrm{GG}(\alpha, \sigma, \tau)$ that leaves W_t marginally $\mathrm{GG}(\alpha, \sigma, \tau)$. For $t = 1, \ldots, T-1$, define

$$C_t = \sum_{i=1}^{\infty} c_{ti}\delta_{\theta_i} \quad c_{ti}|W_t \sim \mathrm{Poisson}(\phi w_{ti}) \tag{6}$$

where $\phi > 0$ is a parameter tuning the correlation. Given C_t, the measure W_{t+1} is then constructed as a combination of masses defined by C_t and GG innovation:

$$W_{t+1} = W^*_{t+1} + \sum_{i=1}^{\infty} w^*_{t+1,i}\delta_{\theta_i} \tag{7}$$

with

$$W^*_{t+1} \sim \mathrm{GG}(\alpha, \sigma, \tau + \phi) \tag{8}$$

$$w^*_{t+1,i}|C_t \sim \mathrm{Gamma}(\max(c_{ti} - \sigma, 0), \tau + \phi). \tag{9}$$

By convention, $\mathrm{Gamma}(0, \tau) = \delta_0$, hence $w^*_{t+1,i} = 0$ if $c_{ti} = 0$ else $w^*_{t+1,i} > 0$, Because the conditional laws of $W_{t+1}|C_t$ coincide with that of $W_t|C_t$ [37, 22, 23], the construction guarantees that W_{t+1} has the same marginal distribution as W_t, i.e., they are both distributed as $\mathrm{GG}(\alpha, \sigma, \tau)$. Moreover, as proved in Section A of the Appendix, the conditional mean of W_{t+1} given $W_t = \sum_i w_{ti}\delta_{\theta_i}$ has the form

$$E[W_{t+1}|W_t] = \left(\frac{\tau}{\tau + \phi}\right)^{1-\sigma} E[W_t] + \frac{1}{\tau + \phi}\sum_{i=1}^{\infty}[\phi w_{ti} - \sigma(1 - e^{-\phi w_{ti}})]\delta_{\theta_i} \tag{10}$$

In the gamma process case ($\sigma = 0$), the above expression reduces to

$$E[W_{t+1}|W_t] = \frac{\tau}{\tau + \phi} E[W_t] + \frac{\phi}{\tau + \phi} W_t.$$

3.3 Summary of the model's hyperparameters

The model is parameterised by $(\alpha, \sigma, \tau, \phi)$, where:

- α tunes the overall size of the networks, with a larger value of α corresponding to larger networks.
- σ controls the sparsity and power-law properties of the graph, as will be shown in Section 4. In Figure 2 we see that different values of σ give rise to different power-law degree distributions.
- τ induces an exponential tilting of large degrees in the degree distribution.
- ϕ tunes the correlation of the sociabilities of each node over time. As we see in Figure 3, larger values correspond to higher correlation and smoother evolution of the weights.

4 Sparsity and power-law properties of the model

By construction, the interactions at time t, n_{tij}, are drawn from the same (static) model as in [6], using a generalised gamma process for the Lévy intensity, and so applying Proposition 18 in [7] we obtain the following asymptotic properties

Proposition 1. *Let $N_{t,\alpha}$ be the number of active nodes at time t, $N^{(e)}_{t,\alpha} = \sum_{i \leq j} z_{tij}$ be the number of edges and $N_{t,\alpha,j}$ the number of nodes of degree j in the graph at time t, then as α tends to infinity, almost surely, we have for any t: if $\sigma > 0$, $N^{(e)}_{t,\alpha} \asymp N^{2/(1+\sigma)}_{t,\alpha}$, if $\sigma = 0$, $N^{(e)}_{t,\alpha} \asymp N^2_{t,\alpha}/\log^2(N_{t,\alpha})$ and if $\sigma < 0$, $N^{(e)}_{t,\alpha} \asymp N^2_{t,\alpha}$. Also, almost surely, for any $t \geq 1$ and $j \geq 1$, if $\sigma \in (0, 1)$,*

$$\frac{N_{t,\alpha,j}}{N_{t,\alpha}} \to p_j, \quad p_j = \frac{\sigma \Gamma(j - \sigma)}{j! \Gamma(1 - \sigma)},$$

while if $\sigma \leq 0$, $N_{t,\alpha,j}/N_{t,\alpha} \to 0$ for all $j \geq 1$.

Hence the graphs are sparse if $\sigma \geq 0$ and dense if $\sigma < 0$.

5 Approximate inference

5.1 Finite-dimensional approximation

Performing exact inference using this model is quite challenging, and we consider instead an approximate inference method, using a finite-dimensional approximation to the GG prior, introduced by [28]. This approximation gives rise

to a particularly simple conjugate construction, enabling posterior inference to be performed.

Let $\text{BFRY}(\eta, \tau, \sigma)$ denote a (scaled and exponentially tilted) BFRY random variable[4] on $(0, \infty)$ with probability density function

$$g_{\eta,\tau,\sigma}(w) = \frac{\sigma w^{-1-\sigma} e^{-\tau w} \left(1 - e^{-(\sigma/\eta)^{1/\sigma} w}\right)}{\Gamma(1-\sigma)\left\{\left(\tau + (\sigma/\eta)^{1/\sigma}\right)^{\sigma} - \tau^{\sigma}\right\}}$$

with parameters $\sigma \in (0, 1)$, $\tau > 0$ and $\eta > 0$.

At time 1, consider the finite-dimensional measure

$$W_1 = \sum_{i=1}^{K} w_{1i} \delta_{\theta_i}$$

where $w_{1i} \sim \text{BFRY}(\alpha/K, \tau, \sigma)$ and $K < \infty$ is the truncation level. As shown by [28], for $\sigma \in (0, 1)$

$$W_1 \xrightarrow{d} \text{GG}(\alpha, \sigma, \tau)$$

as the truncation level K tends to infinity. By this we mean that, if $W \sim GG(\alpha, \sigma, \tau)$, then

$$\lim_{K \to \infty} \mathcal{L}_f(W') = \mathcal{L}_f(W)$$

for an arbitrary measurable and positive f, where $\mathcal{L}_f(W) \coloneqq \mathbb{E}\left[e^{-W(f)}\right]$ is the Laplace functional of W as defined by [28].

To use this finite approximation with our dynamic model, we consider Poisson latent variables as in Equation (6). The measure W_{t+1} is then constructed as:

$$\begin{aligned} W_{t+1} &= \sum_{i=1}^{K} w_{t+1,i} \delta_{\theta_i} \\ w_{t+1,i} | C_t &\sim \text{BFRY}\left(\alpha'_t/K, \tau + \phi, \sigma - c_{ti}\right). \end{aligned}$$

where

$$\alpha'_t = K(\sigma - c_{ti})(\sigma K/\alpha)^{\frac{c_{ti}-\sigma}{\sigma}}.$$

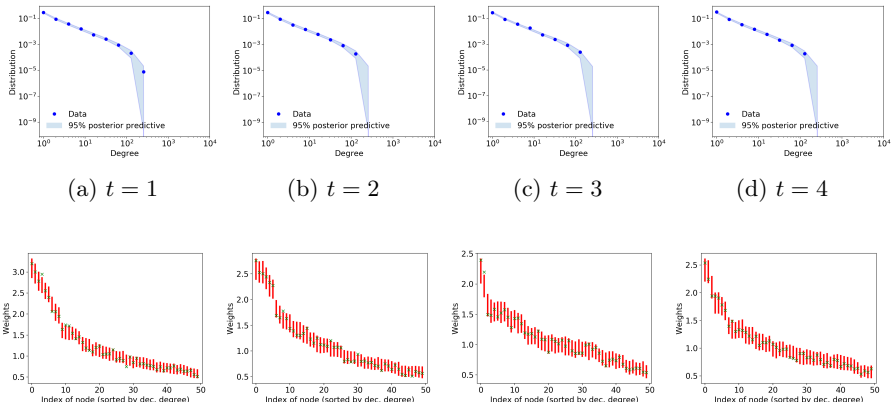
This construction mirrors that of Section 3.2, with the key difference being that we now obtain a stationary $\text{BFRY}(\alpha/K, \tau, \sigma)$ distribution for the w_{ti}. We can easily see this by noting that if

$$\begin{aligned} w_{ti} &\sim \text{BFRY}(\alpha/K, \tau, \sigma) \\ c_{ti} | w_{ti} &\sim \text{Poisson}(\phi w_{ti}) \end{aligned}$$

then

$$p(w_{ti} | c_{ti}) \propto w_{ti}^{-1-\sigma+c_{ti}} e^{-(\tau+\phi) w_{ti}} \left(1 - e^{-(\sigma K/\alpha)^{1/\sigma} w_{ti}}\right) \tag{11}$$

[4] The name was coined by [11] after [2].



(a) $t = 1$ (b) $t = 2$ (c) $t = 3$ (d) $t = 4$

(e) $t = 1$ (f) $t = 2$ (g) $t = 3$ (h) $t = 4$

Fig. 4: (Top row) Posterior Predictive Degree Distribution and (bottom row) 95% credible intervals for sociabilites of the nodes with the highest degrees, for a network simulated from the GG model. True weights are represented by a green cross.

which we then recognise as a BFRY $(\alpha'_t/K, \tau + \phi, \sigma - c_{ti})$ with α'_t as above.

Thus, the use of a finite-dimensional approximation BFRY random variables gives us the simple conjugate construction that we desired. The reason that we introduce this non-standard distribution is that, as far as we know, it is not possible to approximate the generalised gamma process using a finite measure with i.i.d. gamma random weights. In the specific case $\sigma = 0$, we could use the simpler finite approximation using $Gamma(\alpha/K, \tau)$ random variables. However, this would preclude us from modelling networks with power-law degree distributions, as discussed previously.

5.2 Posterior Inference Algorithm

In order to perform posterior inference with the approximate method, we use a Gibbs sampler. We introduce auxiliary variables $\{u_{ti}\}_{i=1}^{K}, t = 1, \ldots, T$ following a truncated exponential distribution. The overall sampler is as follows (see details in Section B of the Appendix):

1. Update the weights w_{ti} given the rest using Hamiltonian Monte Carlo (HMC).
2. Update the latent c_{ti} given the rest using Metropolis Hastings (MH).
3. Update the hyperparameters α, σ, ϕ and τ and the latent variables u_{ti} given the rest

For the hyperparameters, we place gamma priors on α, τ and ϕ, and a Beta prior on σ.

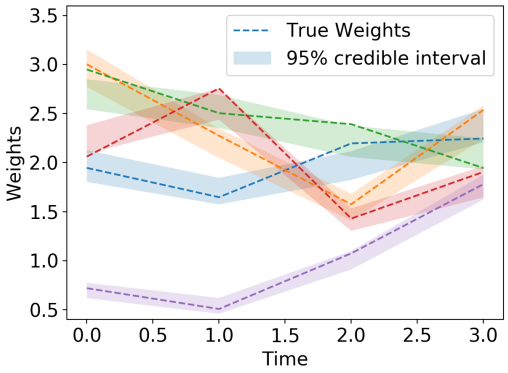


Fig. 5: Evolution of weights of high degree nodes for the network simulated from the GG model. The dotted line shows the true value of the weights in each case.

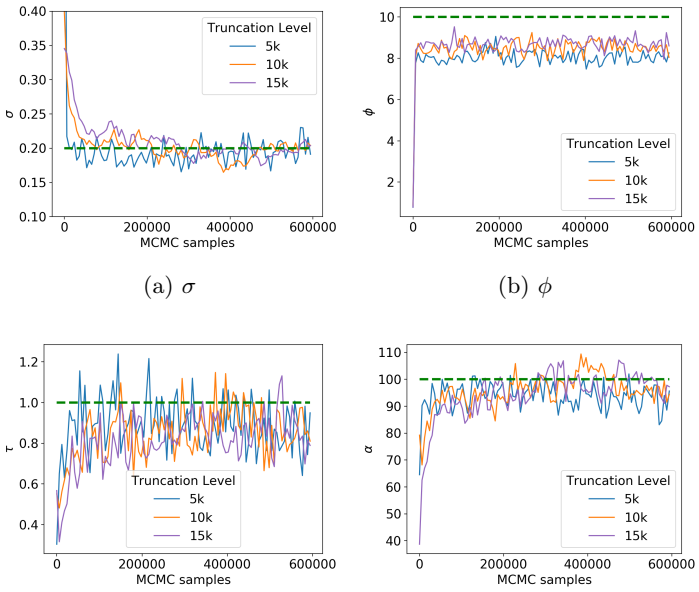
6 Experiments

6.1 Simulated Data

In order to assess the approximate inference scheme, we first consider a synthetic dataset. We simulate a network with $T = 4$, $\alpha = 100$, $\sigma = 0.2$, $\tau = 1$, $\phi = 10$ from the exact model (see Section 3), and then estimate it using our approximate inference scheme described in Section 5. The generated network has $3,096$ nodes and $101,571$ edges. We run an MCMC chain of length $600,000$, with the first $300,000$ samples discarded as burn-in. We set the truncation threshold to $K = 15,000$.

The approximate inference scheme estimates the model's parameters well. This can be seen in terms of the fit of the posterior predictive degree distribution to the empirical as seen in Figures 4a-4d, and the coverage of the credible intervals for the weights, as we see in Figures 4e-4h for the 50 nodes with the highest degree, and in Figure 5 for the evolution over time of some high degree nodes.

To see the influence of the truncation level K, we rerun the algorithm for different truncation levels $K = 5,000$ and $K = 10,000$. Trace plots of the hyperparameters are shown in Figure 6. The approximate posterior distributions are quantitatively similar, but we see that increasing the value of K leads to a more correlated Markov chain. However, we note that ϕ is slightly under-estimated in our model, and this problem is more severe the lower the truncation level is taken to be. Further comparison is given in Section C of the Appendix.



(a) σ (b) ϕ

(c) τ (d) α

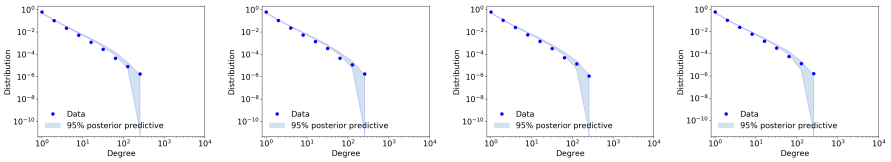
Fig. 6: Trace plots for (a) σ, (b) ϕ, (c) τ and (b) α for different truncation levels, when fitting to a network simulated from the GG model

6.2 Real Data

We illustrate the use of our model on two more dynamic network datasets: the Reddit Hyperlink Network [27][5] and the Reuters Terror dataset[6].

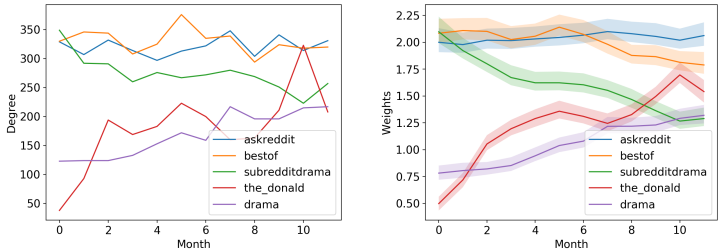
Reddit Hyperlink Network The Reddit hyperlink network represents hyperlink connections between subreddits (communities on Reddit) over a period of $T = 12$ consecutive months in 2016. Nodes are subreddits (communities) and the symmetric edges represent hyperlinks originating in a post in one community and linking to a post in another community. The network has $N = 28,810$ nodes and $388,574$ interactions. The observations here are hyperlinks between the pair of subreddits i, j at time t. The dataset has been made symmetric by placing an edge between nodes i and j if there is a hyperlink between them in either direction. We also assume that there are no loops in the network, that is $n_{tij} = 0$ for $i = j$. We run the Gibbs sampler $400,000$ samples, with the first $200,000$ discarded as burn in. In this case, we choose a truncation level of $K = 40,000$.

[5] `https://snap.stanford.edu/data/soc-RedditHyperlinks.html`
[6] `http://vlado.fmf.uni-lj.si/pub/networks/data/CRA/terror.htm`



(a) $t = 2$ (b) $t = 6$ (c) $t = 10$ (d) $t = 12$

Fig. 7: Posterior Predictive Degree Distribution over time for the Reddit hyperlink network



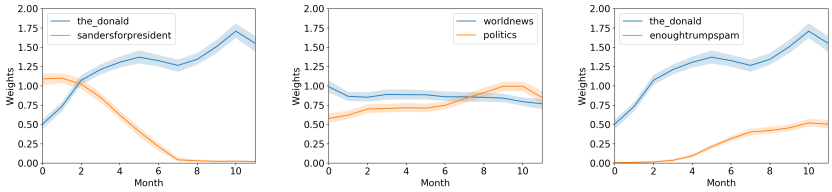
(a) Degree evolution of high degree nodes (b) Weight evolution of high degree nodes

Fig. 8: Evolution of (a) degrees and (b) weights of high degree nodes for the Reddit hyperlink network

From Figure 7 we see that our model is capturing the empirical degree distribution well. Furthermore, in Figure 8 we see that the model is able to capture the evolution of weights associated with each subreddit in a fashion that agrees with the observed frequency of interactions. The high degree nodes here are interpretable as either communities with a very large number of followers - such as "askreddit" or "bestof", or communities which frequently link to others - such as "drama" or "subredditdrama".

In particular, we see in Figure 9a that the weights of the controversial but popular political subreddit "The Donald" increases to a peak in November, corresponding to the 2016 U.S. presidential election. Conversely, the weights of the subreddit "Sandersforpresident" decrease as the year goes on, corresponding to the end of Senator Bernie Sanders' presidential campaign, a trend that again agrees with the evolution of the corresponding observed degrees.

Reuters Terror Dataset The final dataset we consider is the Reuters terror news network dataset. It is based on all stories released during $T = 7$ consecutive weeks (the original data was day-by-day, but was shortened and collated for our purposes) by the Reuters news agency concerning the 09/11/01 attack on the

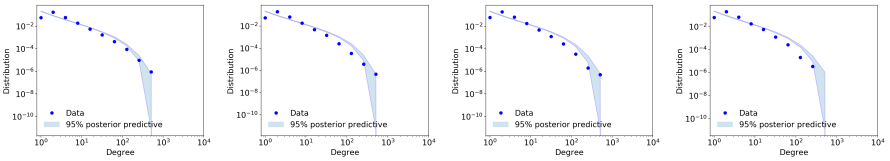


(a) Weights evolution of the subreddits 'the_donald' and 'sandersforpresident'

(b) Weights evolution of the subreddits 'worldnews' and 'politics'

(c) Weights evolution of the subreddits 'the_donald' and 'enoughtrumpspam'

Fig. 9: Evolution of the weights, for the Reddit hyperlink network.



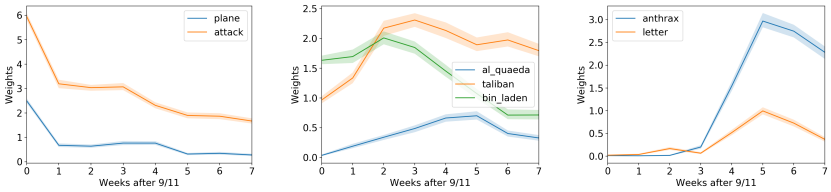
(a) $t = 1$ (b) $t = 3$ (c) $t = 5$ (d) $t = 7$

Fig. 10: Posterior Predictive Degree Distribution for the Reuters terror dataset

U.S.. Nodes are words and edges represent co-occurence of words in a sentence, in news. The network has $N = 13,332$ nodes (different words) and $473,382$ interactions. The observations here are the frequency of co-occurence between the pair of words i, j at time t. We assume that there are no loops in the network, that is $n_{tij} = 0$ for $i = j$. We run the Gibbs sampler with $200,000$ samples, with the first $100,000$ discarded as burn in. In this case, we choose a truncation level of $K = 20,000$.

Figure 10 suggests that the empirical degree distribution does not follow a power law distribution, and our model therefore provides a moderate fit to the empirical degree distribution. The model is however able to capture the evolution of the popularity of the different words, as shown in Figure 11a. For example, the weights of the words "plane" and "attack" decrease over time after 9/11, while the words "letter" and "anthrax" show a peak a few weeks after the attack. These correspond to the anthrax attacks that occurred over several weeks starting a week after 9/11.

Due to the empirical degree distribution not following a power-law, the estimated value of σ is very close to 0, which causes a slow convergence of the MCMC algorithm.



(a) Weights evolution of the words 'plane' and 'attack'

(b) Weights evolution of the words 'al quaeda', 'taliban' and 'bin laden'

(c) Weights evolution of the words 'anthrax' and 'letter'

Fig. 11: Evolution of the weights, for the Reuters terror dataset

7 Discussion and Extensions

A lot of work has been done on modelling dynamic networks; we restrict ourselves to focussing on Bayesian approaches. Much of this work has centred around extending static models. For example, [43] and [12] extend the stochastic block model to the dynamic setting by allowing for parameters that evolve over time. There has also been work on extending the mixed membership stochastic blockmodel [14, 19, 42], the infinite relational model [34] and the latent feature relational model [13, 17, 25]. The problem inherent in these models is that they lead to networks which are dense almost surely [1, 20], a property considered unrealistic for many real-world networks [35] such as social and internet networks.

In order to build models for sparse dynamic networks, [15] build on the framework of edge-exchangeable networks [5, 9, 41], in which graphs are constructed based on an infinitely exchangeable sequence of edges. As in our case, this framework allows for sparse networks with power-law degree distributions. This work, along with others in this framework [34, 16], utilises the mixture of Dirichlet network distributions (MDND) to introduce struture in the networks.

Conversely, our works builds on a different notion of exchangeability [24, 6]. Within this framework, [30] use mutually-exciting Hawkes processes to model temporal interaction data. The difference between our work and theirs is that the sociabilities of the nodes are constant throughout time, with the time evolving element driven by previous interactions via the Hawkes process. Their work also builds on that of [38], incorporating community structure to the network. Exploring how communities appear, evolve and merge could have many practical uses. Thus, expanding our model to capture both evolving node popularity and dynamically changing community structure could be a useful extension.

Furthermore, our model assumes that the time between observations of the network is constant. If this is not the case, it may be helpful to use a continuous-time version of our model. This could be done by considering a birth-death process for the interactions between nodes, where each interaction has a certain lifetime distribution. The continuously evolving node sociabilities could then by described by the Dawson-Watanabe superprocess [40, 10].

Acknowledgements

C. Naik was supported by the Engineering and Physical Sciences Research Council and Medical Research Council [award reference 1930478]. F. Caron was supported by the Engineering and Physical Sciences Council under grant EP/P026753/1. J. Rousseau was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 834175). K. Palla was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617411.

Bibliography

[1] Aldous, D.J.: Representations for partially exchangeable arrays of random variables. Journal of Multivariate Analysis **11**(4), 581–598 (1981)

[2] Bertoin, J., Fujita, T., Roynette, B., Yor, M.: On a particular class of self-decomposable random variables : the durations of Bessel excursions straddling independent exponential times. Probability and Mathematical Statistics **26**(2), 315–366 (2006)

[3] Borgs, C., Chayes, J., Cohn, H., Holden, N.: Sparse exchangeable graphs and their limits via graphon processes. The Journal of Machine Learning Research **18**(1), 1–71 (2018)

[4] Brix, A.: Generalized gamma measures and shot-noise Cox processes. Advances in Applied Probability **31**(4), 929–953 (1999)

[5] Cai, D., Campbell, T., Broderick, T.: Edge-exchangeable graphs and sparsity. In: Advances in Neural Information Processing Systems. pp. 4249–4257 (2016)

[6] Caron, F., Fox, E.: Sparse graphs using exchangeable random measures. Journal of the Royal Statistical Society B **79**, 1–44 (2017), part 5

[7] Caron, F., Panero, F., Rousseau, J.: On sparsity, power-law and clustering properties of graphex processes. arXiv pp. arXiv–1708 (2017)

[8] Caron, F., Teh, Y.W.: Bayesian nonparametric models for ranked data. In: NIPS (2012)

[9] Crane, H., Dempsey, W.: Edge exchangeable models for interaction networks. Journal of the American Statistical Association **113**(523), 1311–1326 (2018)

[10] Dawson, D.A.: Stochastic evolution equations and related measure processes. Journal of Multivariate Analysis **5**(1), 1–52 (1975)

[11] Devroye, L., James, L.F.: On simulation and properties of the stable law. Statistical Methods and Applications **23**(3), 307–343 (2014)

[12] Durante, D., Dunson, D.: Bayesian logistic gaussian process models for dynamic networks. In: AISTATS. pp. 194–201 (2014)

[13] Foulds, J., DuBois, C., Asuncion, A., Butts, C., Smyth, P.: A dynamic relational infinite feature model for longitudinal social networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 287–295 (2011)

[14] Fu, W., Song, L., Xing, E.P.: Dynamic mixed membership blockmodel for evolving networks. In: Proceedings of the 26th annual international conference on machine learning. pp. 329–336 (2009)

[15] Ghalebi, E., Mahyar, H., Grosu, R., Taylor, G.W., Williamson, S.A.: A nonparametric bayesian model for sparse temporal multigraphs. arXiv preprint arXiv:1910.05098 (2019)

[16] Ghalebi, E., Mirzasoleiman, B., Grosu, R., Leskovec, J.: Dynamic network model from partial observations. In: Advances in Neural Information Processing Systems. pp. 9862–9872 (2018)

[17] Heaukulani, C., Ghahramani, Z.: Dynamic probabilistic models for latent feature propagation in social networks. In: International Conference on Machine Learning. pp. 275–283 (2013)
[18] Herlau, T.: Completely random measures for modelling block-structured sparse networks. arXiv preprint arXiv:1507.02925 (2015)
[19] Ho, Q., Song, L., Xing, E.: Evolving cluster mixed-membership blockmodel for time-evolving networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 342–350 (2011)
[20] Hoover, D.N.: Relations on probability spaces and arrays of random variables. Preprint, Institute for Advanced Study, Princeton, NJ (1979)
[21] Hougaard, P.: Survival models for heterogeneous populations derived from stable distributions. Biometrika **73**(2), 387–396 (1986)
[22] James, L.F.: Poisson process partition calculus with applications to exchangeable models and bayesian nonparametrics. arXiv preprint math/0205093 (2002)
[23] James, L.F., Lijoi, A., Prünster, I.: Posterior analysis for normalized random measures with independent increments. Scandinavian Journal of Statistics **36**(1), 76–97 (2009)
[24] Kallenberg, O.: Exchangeable random measures in the plane. Journal of Theoretical Probability **3**(1), 81–136 (1990)
[25] Kim, M., Leskovec, J.: Nonparametric multi-group membership model for dynamic networks. In: Advances in neural information processing systems. pp. 1385–1393 (2013)
[26] Kingman, J.: Completely random measures. Pacific Journal of Mathematics **21**(1), 59–78 (1967)
[27] Kumar, S., Hamilton, W.L., Leskovec, J., Jurafsky, D.: Community interaction and conflict on the web. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web. pp. 933–943. International World Wide Web Conferences Steering Committee (2018)
[28] Lee, J., James, L.F., Choi, S.: Finite-dimensional bfry priors and variational bayesian inference for power law models. In: Advances in Neural Information Processing Systems. pp. 3162–3170 (2016)
[29] Lijoi, A., Mena, R.H., Prünster, I.: Controlling the reinforcement in Bayesian non-parametric mixture models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **69**(4), 715–740 (2007)
[30] Miscouridou, X., Caron, F., Teh, Y.W.: Modelling sparsity, heterogeneity, reciprocity and community structure in temporal interaction data. In: Advances in Neural Information Processing Systems. pp. 2343–2352 (2018)
[31] Naik, C., Caron, F., Rousseau, J.: Sparse networks with core-periphery structure. Electronic Journal of Statistics **15**(1), 1814–1868 (2021)
[32] Neal, R.: Markov chain sampling methods for Dirichlet process mixture models. Journal of computational and graphical statistics pp. 249–265 (2000)
[33] Newman, M.: Networks: an introduction. OUP Oxford (2009)
[34] Ng, Y.C., Silva, R.: A dynamic edge exchangeable model for sparse temporal networks. arXiv preprint arXiv:1710.04008 (2017)

[35] Orbanz, P., Roy, D.M.: Bayesian models of graphs, arrays and other exchangeable random structures. IEEE Trans. Pattern Anal. Mach. Intelligence (PAMI) **37**(2), 437–461 (2015)
[36] Pitt, M.K., Walker, S.G.: Constructing stationary time series models using auxiliary variables with applications. Journal of the American Statistical Association **100**(470), 554–564 (2005)
[37] Prünster, I.: Random probability measures derived from increasing additive processes and their application to Bayesian statistics. Ph.D. thesis, University of Pavia (2002)
[38] Todeschini, A., X., M., Caron, F.: Exchangeable random measures for sparse and modular graphs with overlapping communities. Journal of the Royal Statistical Society B (2020)
[39] Veitch, V., Roy, D.M.: The class of random graphs arising from exchangeable random measures. arXiv preprint arXiv:1512.03099 (2015)
[40] Watanabe, S.: A limit theorem of branching processes and continuous state branching processes. Journal of Mathematics of Kyoto University **8**(1), 141–167 (1968)
[41] Williamson, S.A.: Nonparametric network models for link prediction. The Journal of Machine Learning Research **17**(1), 7102–7121 (2016)
[42] Xing, E.P., Fu, W., Song, L., et al.: A state-space mixed membership blockmodel for dynamic network tomography. The Annals of Applied Statistics **4**(2), 535–566 (2010)
[43] Xu, K., Hero, A.O.: Dynamic stochastic blockmodels for time-evolving social networks. Selected Topics in Signal Processing, IEEE Journal of **8**(4), 552–562 (2014)

A Proofs

A.1 Proof of Equation (10)

$$\begin{aligned}
&E[W_{t+1}|W_t] \\
&= E[E[W_{t+1}|C_t]|W_t] \\
&= E[W^*_{t+1}] + \frac{1}{\tau+\phi}\sum_{i=1}^{\infty} E[\max(c_{ti}-\sigma,0)|W_t]\delta_{\theta_i} \\
&= E[W^*_{t+1}] + \frac{1}{\tau+\phi}\sum_{i=1}^{\infty} [\phi w_{ti} - \sigma(1-e^{-\phi w_{ti}})]\delta_{\theta_i}
\end{aligned}$$

Finally, note that, for any measurable set $A \subseteq [0,\alpha]$, using Campbell's theorem,

$$E[W_{t+1}(A)] = \lambda(A)\int_0^{\infty} w\rho(w)dw = \lambda(A)\tau^{\sigma-1}$$

and similarly

$$\begin{aligned}
E[W^*_{t+1}(A)] &= \lambda(A)(\tau+\phi)^{\sigma-1} \\
&= \left(\frac{\tau}{\tau+\phi}\right)^{1-\sigma} E[W_{t+1}(A)]
\end{aligned}$$

where λ denotes the Lebesgue measure. Thus,

$$\begin{aligned}
E[W_{t+1}|W_t] = &\left(\frac{\tau}{\tau+\phi}\right)^{1-\sigma} E[W_t] \\
&+ \frac{1}{\tau+\phi}\sum_{i=1}^{\infty} [\phi w_{ti} - \sigma(1-e^{-\phi w_{ti}})]\delta_{\theta_i}
\end{aligned} \tag{12}$$

B MCMC algorithm details

For $t = 1,\ldots,T$, let $\mathbf{n}_t = (n_{tij})_{1\le i,j\le K}$, $\mathbf{w}_t = (w_{ti})_{i=1,\ldots,K}$, $\mathbf{u}_t = (u_{ti})_{i=1,\ldots,K}$ and $\mathbf{c}_t = (c_{ti})_{i=1,\ldots,K}$.

B.1 Conditional distribution for the interactions n_{tij}

The conditional distribution of $(\mathbf{n}_t)_{t=1,\ldots,T}$ given the other variables can be found by recalling Equation (4) in the main article. We have

$$p((\mathbf{n}_t)_{t=1,\ldots,T} \mid \text{rest}) = \prod_{t=1}^{T} p(\mathbf{n}_t \mid \mathbf{w}_t)$$

where

$$p(\mathbf{n}_t \mid \mathbf{w}_t) = \prod_{1\le i<j\le K} \frac{(2w_{ti}w_{tj})^{n_{tij}} e^{-2w_{ti}w_{tj}}}{n_{tij}!} \prod_{i=1}^{K} \frac{(w_{ti}^2)^{n_{tii}} e^{-w_{ti}^2}}{n_{tii}!}$$
$$\propto \Big[\prod_{i=1}^{K} w_{ti}^{m_{ti}}\Big] e^{-(\sum_{i=1}^{K} w_{ti})^2}$$

where $m_{ti} = n_{tii} + \sum_{j=1}^{K} n_{tij}$.

B.2 Update of u_{ti} and w_{ti}

The auxiliary variables u_{ti} are introduced in order to help with numerical instability problems with the update of the weights w_{ti}. These generally stem from the $\left(1 - e^{-(\sigma K/\alpha)^{1/\sigma} w_{ti}}\right)$ term in the BFRY distribution, which can be unstable when $(\sigma K/\alpha)^{1/\sigma}$ is large. Noting that the truncation level K is a fixed value that we choose, we denote this term by

$$t_{\alpha,\sigma} = (\sigma K/\alpha)^{1/\sigma} \tag{13}$$

in everything that follows.

When defining u_{ti}, we want to cancel this problematic term out from the distribution of w_{ti}, and thus we sample u_{ti}:

$$u_{ti} \mid w_{ti} \sim \mathrm{tExp}(w_{ti}, t_{\alpha,\sigma}) \tag{14}$$

where $\mathrm{tExp}(\lambda, a)$ denotes an Exponential distribution with rate parameter λ, truncated on $[0, a]$. The density function of $u_{ti}|w_{ti}$ is thus given by:

$$p(u_{ti}|w_{ti}) = \frac{w_{ti} e^{-u_{ti}w_{ti}}}{(1 - e^{-t_{\alpha,\sigma}w_{ti}})}, \qquad 0 \le u_{ti} \le t_{\alpha,\sigma} \tag{15}$$

We can sample u_{ti} directly using inverse transform sampling. However, due to non-conjugacy we cannot sample directly from the posterior of w_{ti}. For that reason we use Hamiltonian Monte Carlo (HMC) [32]. For a given value of t, the posterior of $\mathbf{w}_t$ is given by:

$$\begin{aligned}
&p(\mathbf{w}_t|\mathbf{n}_t, \mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{u}_t) \\
&\propto p(\mathbf{n}_t|\mathbf{w}_t) p(\mathbf{w}_t|\mathbf{c}_t, \mathbf{c}_{t-1}) p(\mathbf{u}_t|\mathbf{w}_t) \\
&\propto \Big[\prod_{i=1}^{K} w_{ti}^{m_i}\Big] e^{-(\sum_{i=1}^{K} w_{ti})^2} \Big[\prod_{i=1}^{K} w_{ti}^{c_{ti}+\mathbb{1}_{t>1}c_{t-1i}-1-\sigma} e^{-(\phi+\mathbb{1}_{t>1}\phi+\tau)w_{ti}} \left(1 - e^{-t_{\alpha,\sigma}w_{ti}}\right)\Big] \\
&\quad \times \Big[\prod_{i=1}^{K} \frac{w_{ti} e^{-u_{ti}w_{ti}}}{(1 - e^{-t_{\alpha,\sigma}w_{ti}})}\Big] \\
&\propto \Big[\prod_{i=1}^{K} w_{ti}^{c_{ti}+\mathbb{1}_{t>1}c_{t-1i}+1+m_i-1-\sigma} e^{-(\phi+\mathbb{1}_{t>1}\phi+\tau+u_{ti})w_{ti}}\Big] e^{-(\sum_{i=1}^{K} w_{ti})^2}
\end{aligned}$$

where some of the terms only appear for $\mathbb{1}_{t>1}$ because $\mathbf{c}_0$ does not exist, and thus $p(w_{1i}|c_{1i}, c_{0,i}) = p(w_{1i}|c_{1i})$. Here, we used that

$$p(\mathbf{w}_t|\mathbf{c}_t, \mathbf{c}_{t-1}) \propto \prod_{i=1}^{K} \left[\text{BFRY}\left(\alpha_t''/K, \tau + 2\phi, \sigma - c_{ti} - c_{t-1i}\right) \right] \tag{16}$$

while noting that the form of α_t'' is defined such that

$$t_{\alpha,\sigma} = (\sigma K/\alpha)^{1/\sigma} = \left(\frac{(\sigma - c_{ti} - c_{t-1i})K}{\alpha_t''} \right)^{\frac{1}{\sigma - c_{ti} - c_{t-1i}}}$$

We use change of variables $y_{ti} = \log w_{ti}$. The HMC algorithm requires computing the gradient of the log-posterior which is:

$$\begin{aligned} \frac{\partial \log p(\mathbf{y}_t|\mathbf{n}_t, \mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{u}_t)}{dy_{ti}} &= (c_{ti} + \mathbb{1}_{t>1} c_{t-1i} + 1 + m_{ti} - \sigma) \\ &\quad - w_{ti} \left(\phi + \mathbb{1}_{t>1}\phi + \tau + u_{ti} + 2\sum_{j=1}^{K} w_{tj} \right) \end{aligned} \tag{17}$$

The algorithm then proceeds as follows, for each $t = 1, \ldots, T$:

1. Sample momentum variables $\mathbf{p}_t = (p_{ti})_{i=1,\ldots,K}$ as:

$$p_{ti} \overset{iid}{\sim} \mathcal{N}(0, 1) \quad i = 1, \ldots, K$$

2. Simulate L steps of the discretized Hamiltonian via

$$\begin{aligned} \log \tilde{\mathbf{w}}_t^{(0)} &= \log \mathbf{w}_t \\ \tilde{p}_{ti} &= p_{ti} + \frac{\varepsilon}{2} \frac{\partial \log p(\log \mathbf{w}_t|\mathbf{n}_t, \mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{u}_t)}{d(\log w_{ti})} \quad i = 1, \ldots, K \end{aligned}$$

and for $l = 1, \ldots, L-1$

$$\begin{aligned} \log \tilde{\mathbf{w}}_t^{(l)} &= \log \tilde{\mathbf{w}}_t^{(l-1)} + \varepsilon \tilde{\mathbf{p}}_t^{(l-1)} \\ \tilde{p}_{ti}^{(l)} &= \tilde{p}_{ti}^{(l-1)} + \varepsilon \frac{\partial \log p(\log \tilde{\mathbf{w}}_t^{(l)}|\mathbf{n}_t, \mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{u}_t)}{d(\log \tilde{w}_{ti}^{(l)})} \end{aligned}$$

where $\tilde{\mathbf{p}}_t = (\tilde{p}_{ti})_{i=1,\ldots,K}$. Finally, set

$$\begin{aligned} \log \tilde{\mathbf{w}}_t &= \log \tilde{\mathbf{w}}_t^{(L-1)} + \varepsilon \tilde{\mathbf{p}}_t^{(L-1)} \\ \tilde{p}_{ti} &= -\left[\tilde{p}_{ti}^{(L-1)} + \frac{\varepsilon}{2} \frac{\partial \log p(\log \tilde{\mathbf{w}}_t|\mathbf{n}_t, \mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{u}_t)}{d(\log \tilde{w}_{ti})} \right] \quad i = 1, \ldots, K \end{aligned}$$

3. Accept $\tilde{\mathbf{w}}_t$, with probability $a = \min(1, r)$, where r is given by:

$$\begin{aligned} r = &\Big[\prod_{i=1}^{K} \Big(\frac{\tilde{w}_{ti}}{w_{ti}} \Big)^{c_{ti} + \mathbb{1}_{t>1} c_{t-1i} + 1 + m_{ti} - \sigma} \Big] \\ &\times e^{-(\sum_{i=1}^{K} \tilde{w}_{ti})^2 + (\sum_{i=1}^{K} w_{ti})^2 - (\phi + \mathbb{1}_{t>1}\phi + \tau)(\sum_{i=1}^{K} \tilde{w}_{ti} - \sum_{i=1}^{K} w_{ti})} \\ &\times e^{-\sum_{i=1}^{K} u_{ti}\tilde{w}_{ti} - \sum_{i=1}^{K} u_{ti} w_{ti}} e^{-\frac{1}{2}\sum_{i=1}^{K} \tilde{p}_{ti}^2 - p_{ti}^2} \end{aligned} \tag{18}$$

In our case, we take $L = 10$, and tune ε on the first $10,000$ samples in order to achieve an acceptance rate of 0.65.

B.3 Update of c_{ti}

We update $\mathbf{c}_t|\mathbf{w}_t, \mathbf{w}_{t+1}$ for each t. The conditional distribution is:

$$p(c_{tk}|w_{tk}, w_{t+1k}) \propto p(c_{tk}|w_{tk})p(w_{t+1k}|c_{tk}) \tag{19}$$

where

$$p(c_{tk}|w_{tk}) = \text{Poisson}(c_{tk}; \phi w_{tk}) \tag{20}$$
$$p(w_{t+1,k}|c_{tk}, w_{tk}) = \text{BFRY}\left(w_{t+1,k}; \alpha'_t/K, \tau + \phi, \sigma - c_{ti}\right) \tag{21}$$

For each $t = 1, \ldots, T$ and $k = 1, \ldots K$. We use Metropolis-Hastings to sample c_{tk} with a Poisson as the proposal distribution, i.e. $\tilde{c}_{tk} \sim \text{Poisson}(\phi w_k)$. Accept with probability $\min(1, r)$ where

$$r = \frac{p(w_{t+1,k}|\tilde{c}_{tk})}{p(w_{t+1,k}|c_{tk})} = \frac{\text{BFRY}\left(w_{t+1,k}; \tilde{\alpha}'_t/K, \tau + \phi, \sigma - \tilde{c}_{ti}\right)}{\text{BFRY}\left(w_{t+1,k}; \alpha'_t/K, \tau + \phi, \sigma - c_{ti}\right)} \tag{22}$$

where we remember that α'_t depends on c_t.

B.4 Update of α, σ, τ, ϕ

The joint posterior is given by

$$\begin{aligned} p(\alpha, \sigma, \tau, \phi|C, W) \propto & \prod_{k=1}^{K} \left(p(w_{1k}) p(c_{1k}|\phi, w_{1k}) \right) \prod_{t=2}^{T} \left[\prod_{k=1}^{K} \left(p(w_{tk}|c_{t-1k}) p(c_{tk}|\phi, w_{tk}) \right) \right] \\ & \times p(\alpha)p(\sigma)p(\tau)p(\phi) \\ \propto & \prod_{k=1}^{K} \left(\text{BFRY}\left(w_{1k}; \alpha/K, \tau, \sigma\right) \frac{(w_{1k}\phi)^{c_{1k}} e^{-\phi w_{1k}}}{c_{1k}!} \right) \\ & \times \prod_{t=2}^{T} \left[\prod_{k=1}^{K} \left(\text{BFRY}\left(w_{tk}; \alpha'_t/K, \tau + \phi, \sigma - c_{t-1,k}\right) \frac{(w_{tk}\phi)^{c_{tk}} e^{-\phi w_{tk}}}{c_{tk}!} \right) \right] \\ & \times p(\alpha)p(\sigma)p(\tau)p(\phi) \end{aligned} \tag{23}$$

We place Gamma priors on α, τ and ϕ, and a Beta prior on σ:

$$p(\alpha) = \text{Gamma}(\alpha; a_1, a_2) \tag{24}$$
$$p(\sigma) = \text{Beta}(\sigma; s_1, s_2) = \frac{\sigma^{s_1-1}(1-\sigma)^{s_2-1}}{B(s_1, s_2)} \tag{25}$$
$$p(\tau) = \text{Gamma}(\tau; t_1, t_2) \tag{26}$$
$$p(\phi) = \text{Gamma}(\phi; f_1, f_2) \tag{27}$$

We then use a Metropolis Hastings sampler, with proposals given by:

$$q(\tilde{\alpha}|\alpha) = \text{lognormal}(\tilde{\alpha}; \log(\alpha), \sigma_\alpha) = \frac{1}{\sigma_\alpha\sqrt{2\pi}\tilde{\alpha}}\exp\left(-\frac{(\log\tilde{\alpha} - \log\alpha)^2}{2\sigma_\alpha^2}\right) \tag{28}$$

$$q(\tilde{\tau}|\tau) = \text{lognormal}(\tilde{\tau}; \log(\tau), \sigma_\tau) = \frac{1}{\sigma_\tau\sqrt{2\pi}\tilde{\tau}}\exp\left(-\frac{(\log\tilde{\tau} - \log\tau)^2}{2\sigma_\tau^2}\right) \tag{29}$$

$$q(\tilde{\phi}|\phi) = \text{lognormal}\left(\tilde{\phi}; \log(\phi), \sigma_\phi\right) = \frac{1}{\sigma_\phi\sqrt{2\pi}\tilde{\phi}}\exp\left(-\frac{(\log\tilde{\phi} - \log\phi)^2}{2\sigma_\phi^2}\right) \tag{30}$$

$$q(\tilde{\sigma}|\sigma) = \frac{1}{\sigma_\sigma\sqrt{2\pi}\tilde{\sigma}(1-\tilde{\sigma})}\exp\left(-\frac{(\log\frac{\tilde{\sigma}}{1-\tilde{\sigma}} - \log\frac{\sigma}{1-\sigma})^2}{2\sigma_\sigma^2}\right) \tag{31}$$

The parts of the MH ratio involving the priors and proposals of the hyperparameters are given by:

$$\frac{p(\tilde{\alpha})q(\alpha|\tilde{\alpha})}{p(\alpha)q(\tilde{\alpha}|\alpha)} = e^{-a_2(\tilde{\alpha}-\alpha)}\left(\frac{\tilde{\alpha}}{\alpha}\right)^{a_1} \tag{32}$$

$$\frac{p(\tilde{\tau})q(\tau|\tilde{\tau})}{p(\tau)q(\tilde{\tau}|\tau)} = e^{-t_2(\tilde{\tau}-\tau)}\left(\frac{\tilde{\tau}}{\tau}\right)^{t_1} \tag{33}$$

$$\frac{p(\tilde{\phi})q(\phi|\tilde{\phi})}{p(\phi)q(\tilde{\phi}|\phi)} = e^{-f_2(\tilde{\phi}-\phi)}\left(\frac{\tilde{\phi}}{\phi}\right)^{f_1} \tag{34}$$

$$\frac{p(\tilde{\sigma})q(\sigma|\tilde{\sigma})}{p(\sigma)q(\tilde{\sigma}|\sigma)} = \left(\frac{\tilde{\sigma}}{\sigma}\right)^{s_1}\left(\frac{1-\tilde{\sigma}}{1-\sigma}\right)^{s_2} \tag{35}$$

The MH ratio without the priors and proposals on α, σ, τ and ϕ is given by

$$\frac{\prod_{k=1}^{K}\left(\text{BFRY}\left(w_{1k};\tilde{\alpha}/K,\tilde{\tau},\tilde{\sigma}\right)\frac{(w_{1k}\tilde{\phi})^{c_{1k}}e^{-\tilde{\phi}w_{1k}}}{c_{1k}!}\right) \times \prod_{t=2}^{T}\left[\prod_{k=1}^{K}\left(\text{BFRY}\left(w_{tk};\tilde{\alpha}'_t/K,\tilde{\tau}+\tilde{\phi},\tilde{\sigma}-c_{t-1,k}\right)\frac{(w_{tk}\tilde{\phi})^{c_{tk}}e^{-\tilde{\phi}w_{tk}}}{c_{tk}!}\right)\right]}{\prod_{k=1}^{K}\left(\text{BFRY}\left(w_{1k};\alpha/K,\tau,\sigma\right)\frac{(w_{1k}\phi)^{c_{1k}}e^{-\phi w_{1k}}}{c_{1k}!}\right) \times \prod_{t=2}^{T}\left[\prod_{k=1}^{K}\left(\text{BFRY}\left(w_{tk};\alpha'_t/K,\tau+\phi,\sigma-c_{t-1,k}\right)\frac{(w_{tk}\phi)^{c_{tk}}e^{-\phi w_{tk}}}{c_{tk}!}\right)\right]} \tag{36}$$

The logarithm of the ratio is then given by:

$$
\begin{aligned}
&K\left[\log(\tilde{\sigma})-\log(\sigma)\right]+K\left[\log(\Gamma(1-\sigma))-\log(\Gamma(1-\tilde{\sigma}))\right]+\left[-(\tilde{\tau}+\tilde{\phi})+(\tau+\phi)\right]\sum_{k=1}^{K}w_{1k}\\
&+K\left[\log\left(\{(\tau+t_{\alpha,\sigma})^{\sigma}-\tau^{\sigma}\}\right)-\log\left(\left\{(\tilde{\tau}+t_{\tilde{\alpha},\tilde{\sigma}})^{\tilde{\sigma}}-\tilde{\tau}^{\tilde{\sigma}}\right\}\right)\right]\\
&+(\sigma-\tilde{\sigma})\sum_{t=1}^{T}\sum_{k=1}^{K}\log(w_{tk})+\left[\log(\tilde{\phi})-\log\phi)\right]\sum_{t=1}^{T}\sum_{k=1}^{K}c_{tk}\\
&+\sum_{t=1}^{T}\sum_{k=1}^{K}\left[\log\left(1-e^{-t_{\tilde{\alpha},\tilde{\sigma}}w_{tk}}\right)-\log\left(1-e^{-t_{\alpha,\sigma}w_{tk}}\right)\right]\\
&+\sum_{t=2}^{T}\sum_{k=1}^{K}\log\left(\frac{\Gamma(1-\sigma+c_{t-1k})}{\Gamma(1-\tilde{\sigma}+c_{t-1k})}\right)+\left[-(\tilde{\tau}+2\tilde{\phi})+(\tau+2\phi)\right]\sum_{t=2}^{T}\sum_{k=1}^{K}w_{tk}\\
&+\sum_{t=2}^{T}\sum_{k=1}^{K}\log\left(\frac{\tilde{\sigma}-c_{t-1k}}{\sigma-c_{t-1k}}\right)+\sum_{t=2}^{T}\sum_{k=1}^{K}\log\left(\frac{\left\{(\tau+\phi+t_{\alpha,\sigma})^{\sigma-c_{t-1k}}-(\tau+\phi)^{\sigma-c_{t-1k}}\right\}}{\left\{\left(\tilde{\tau}+\tilde{\phi}+t_{\tilde{\alpha},\tilde{\sigma}}\right)^{\tilde{\sigma}-c_{t-1k}}-\left(\tilde{\tau}+\tilde{\phi}\right)^{\tilde{\sigma}-c_{t-1k}}\right\}}\right)
\end{aligned}
\tag{37}
$$

B.5 Log-posterior density

The posterior probability density function, given the latent variables and up to a normalizing constant, thus takes the form:

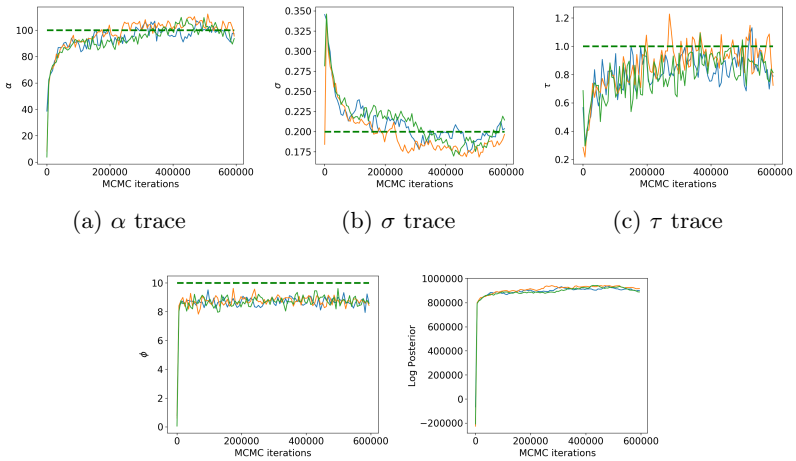
$$
\begin{aligned}
&p\left((w_{tk},c_{tk})_{k=1,\ldots,K,t=1,\ldots,T},\sigma,\tau,\phi,\alpha,\rho\mid(n_{tij})_{1\leq i,j\leq K,t=1,\ldots,T}\right)\\
&\propto\prod_{t=1}^{T}\left[p(\mathbf{n}_t|\mathbf{w}_t)\right]\prod_{k=1}^{K}\left(p(w_{1k})p(c_{1k}|\phi,w_{1k})\right)\prod_{t=2}^{T}\left[\prod_{k=1}^{K}\left(p(w_{tk}|c_{t-1k})p(c_{tk}|\phi,w_{tk})\right)\right]\\
&\quad\times p(\alpha)p(\sigma)p(\tau)p(\phi)\\
&\propto\prod_{t=1}^{T}\left[\left[\prod_{k=1}^{K}w_{tk}^{m_{tk}}\right]e^{-(\sum_{k=1}^{K}w_{tk})^2)}\right]\prod_{k=1}^{K}\left(\mathrm{BFRY}\left(w_{1k};\alpha/K,\tau,\sigma\right)\frac{(w_{1k}\phi)^{c_{1k}}e^{-\phi w_{1k}}}{c_{1k}!}\right)\\
&\quad\times\prod_{t=2}^{T}\left[\prod_{k=1}^{K}\left(\mathrm{BFRY}\left(w_{tk};\alpha'_t/K,\tau+\phi,\sigma-c_{t-1,k}\right)\frac{(w_{tk}\phi)^{c_{tk}}e^{-\phi w_{tk}}}{c_{tk}!}\right)\right]\\
&\quad\times p(\alpha)p(\sigma)p(\tau)p(\phi).
\end{aligned}
\tag{38}
$$

C MCMC plots

In this section we give the MCMC trace plots for the simulated and real data experiments. In each case, the samples are thinned so that we have 100 samples from the posterior.

C.1 Simulated Data

In Figure 12 we see the trace plots for the hyperparameters α, σ, τ and ϕ, as well at the logposterior (up to a constant), for the network simulated from the GG model. We can see that the parameters other than ϕ converge well to their true values, and that in each case the mixing of the three chains is good. Furthermore, we can see that the logposterior has converged to a stable value.



(a) α trace (b) σ trace (c) τ trace

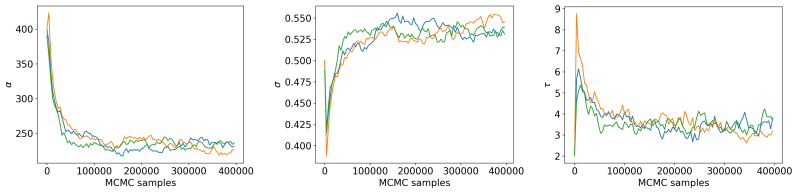
(d) ϕ trace (e) Logposterior trace

Fig. 12: MCMC trace plots of hyperparameters and logposterior for the network simulated from the GG model, with $\alpha = 100$, $\sigma = 0.2$, $\tau = 1$ and $\phi = 10$. The true values in each case are denoted by the dotted green line.

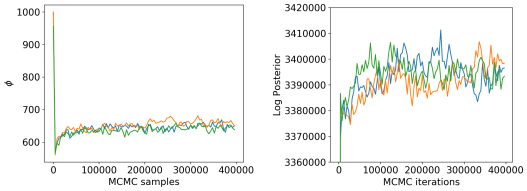
C.2 Real Data

Reddit Hyperlink Network In Figure 13 we see the trace plots for the hyperparameters α, σ, τ and ϕ, as well at the logposterior, for the Reddit hyperlink network. We can see that the parameters and the logposterior have converged to a stable value.

Reuters Terror Dataset In Figure 14 we see the trace plots for the hyperparameters α, σ, τ and ϕ, as well at the logposterior, for the Reuters terror network. As before, we see that the hyperparameters have not yet converged. However, we found that running a longer chain provided similar credible intervals for the weights and degree distribution.

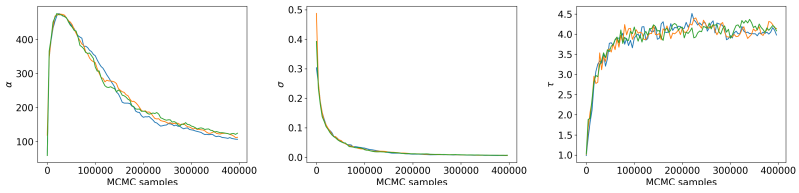


(a) α trace (b) σ trace (c) τ trace

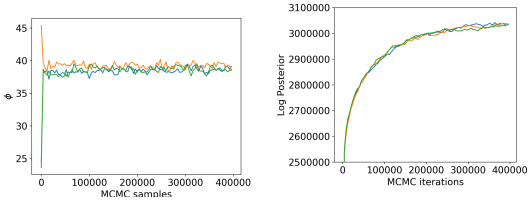


(d) ϕ trace (e) Logposterior trace

Fig. 13: MCMC trace plots of hyperparameters and logposterior for the Reddit hyperlink network.



(a) α trace (b) σ trace (c) τ trace



(d) ϕ trace (e) Logposterior trace

Fig. 14: MCMC trace plots of hyperparameters and logposterior for the Reuters terror network.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Bayesian nonparametrics for sparse dynamic networks
Publication Status	□Published □ Accepted for Publication ■Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Submitted to the European Conference on Machine Learning and Data Mining, 2022

Student Confirmation

Student Name:	Cian Naik		
Contribution to the Paper	I am the first author of this paper. I have contributed to developing the methodology, designing and implementing the inference scheme, running the experiments and writing the paper.		
Signature	Cian Naik	Date	25/03/22

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor François Caron		
Supervisor comments I certify that Cian made a substantial contribution to the publication, as described above.		
Signature	Date	28/03/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

4

Fast Bayesian Coresets via Subsampling and Quasi-Newton Refinement

arXiv:2203.09675v2 [stat.ML] 21 Jul 2022

Fast Bayesian Coresets via Subsampling and Quasi-Newton Refinement

Cian Naik
Department of Statistics
University of Oxford
cian.naik@stats.ox.ac.uk

Judith Rousseau
Department of Statistics
University of Oxford
judith.rousseau@stats.ox.ac.uk

Trevor Campbell
Department of Statistics
University of British Columbia
trevor@stat.ubc.ca

Abstract

Bayesian coresets approximate a posterior distribution by building a small weighted subset of the data points. Any inference procedure that is too computationally expensive to be run on the full posterior can instead be run inexpensively on the coreset, with results that approximate those on the full data. However, current approaches are limited by either a significant run-time or the need for the user to specify a low-cost approximation to the full posterior. We propose a Bayesian coreset construction algorithm that first selects a uniformly random subset of data, and then optimizes the weights using a novel quasi-Newton method. Our algorithm is a simple to implement, black-box method, that does not require the user to specify a low-cost posterior approximation. It is the first to come with a general high-probability bound on the KL divergence of the output coreset posterior. Experiments demonstrate that our method provides significant improvements in coreset quality against alternatives with comparable construction times, with far less storage cost and user input required.

1 Introduction

Bayesian methods are key tools for parameter estimation and uncertainty quantification, but exact inference is rarely possible for complex models. Currently, the gold standard method for approximate inference is Markov chain Monte Carlo (MCMC) [1; 2; 3, Ch. 11,12], which involves simulating a Markov chain whose stationary distribution is the Bayesian posterior. However, modern applications are often concerned with very large datasets; in this setting, MCMC methods typically have a $\Theta(NT)$ complexity—for T samples and dataset size N —which quickly becomes intractable as N increases. Motivated by stochastic methods in variational inference [4, 5], this cost can be reduced by involving only a random subsample of $M \ll N$ data points in each Markov chain iteration [6–13]. This reduces the per-iteration computation time of MCMC, but it can create substantial error in the stationary distribution of the resulting Markov chain and cause slow mixing [11, 14–17].

Bayesian coresets [18–23] provide an alternative for reducing the cost of MCMC and other inference methods. The key idea is that in a large-scale data setting, much of the data is often redundant. In particular, there often exists a *fixed* small, weighted subset of the data—a *coreset*—that suffices to capture the dataset as a whole in some sense [21]. Thus, if one can find such a coreset, it can be used in place of the full dataset in the MCMC algorithm, providing the simplicity, generality, and per-iteration speed of data-subsampled MCMC without the statistical drawbacks. Current state-of-

Preprint. Under review.

the-art Bayesian coreset approaches based on sparse variational inference [21] empirically provide high-quality coresets, but are limited by their significant run-time and lack of theoretical convergence guarantees. Methods based on sparse regression [19, 20, 23] are significantly faster and come with some limited theoretical guarantees, but require a low-cost approximation to the full posterior that is often impractical to find and difficult to tune. Aside from [21], these methods also require $\Theta(N)$ storage, which is problematic in the large-data setting. There is a third class of constructions based on importance sampling [18], but these do not provide reliable posterior approximations in practice.

In this work, we develop a novel coreset construction algorithm that is both faster and easier to tune than the current state of the art, and has theoretical guarantees on the quality of its output. The key insight in our work is that we can split the construction of a coreset into two stages: first we select the data in the coreset via uniform subsampling—an idea developed concurrently by [24, 25]—and then optimize the weights using a few steps of a novel quasi-Newton method. Selecting points uniformly randomly avoids the slow inner-outer loop optimization of [21], and guarantees that the optimally-weighted coreset has a low KL divergence to the posterior (Theorems 4.1 and 4.2). Weighting these uniformly selected points correctly is crucial, and our quasi-Newton method is guaranteed to converge to a point close to the optimally-weighted coreset (Theorem 4.3) in significantly fewer iterations than the method of [21]. Finally, our method is easy to tune and does not require the low-cost posterior approximation of [19, 20]. Our experiments show that the algorithm exhibits significant improvements in coreset quality against alternatives with comparable construction times.

2 Bayesian coresets

The problem we study is as follows. Suppose we are given a target probability density $\pi(\theta)$ for variable $\theta \in \Theta$ that is comprised of N potentials $(f_n(\theta))_{n=1}^N$ and base density $\pi_0(\theta)$,

$$\pi(\theta) \coloneqq \frac{1}{Z} \exp\left(\sum_{n=1}^N f_n(\theta)\right) \pi_0(\theta), \tag{1}$$

where Z is the normalizing constant. This setup corresponds to a Bayesian statistical model with prior π_0 and i.i.d. data X_n conditioned on θ, where $f_n(\theta) = \log p(X_n|\theta)$.

Computing expectations under π exactly is often intractable. MCMC methods can be employed here, but typically have a computational complexity of $\Theta(NT)$ to obtain T samples, since $\sum_n f_n(\theta)$ needs to be evaluated at each step. Bayesian coresets [18] offer an alternative, by constructing a small, weighted subset of the dataset, on which any MCMC algorithm can then be run.

To do this, we find a set of weights $w \in \mathbb{R}^N_{\geq 0}$ corresponding to each data point, with the constraint that only $M \ll N$ of these are nonzero, i.e. $\|w\|_0 \coloneqq \sum_{n=1}^N \mathbb{1}_{w_n > 0} \leq M$. The weighted subset of points corresponding to the strictly positive weights is called a *coreset*, and we can then create the coreset posterior approximation,

$$\pi_w(\theta) \coloneqq \frac{1}{Z(w)} \exp\left(\sum_{n=1}^N w_n f_n(\theta)\right) \pi_0(\theta), \tag{2}$$

where $Z(w)$ is the new normalizing constant, and $\pi_1 = \pi$ corresponds to the full posterior. Running MCMC on this approximation has complexity $\Theta(MT)$, a considerable speedup if $M \ll N$.

Constructing a sparse set of weights w so that π_w is as close to π as possible is the key challenge here. Methods based on importance sampling [18] and sparse regression [19, 20, 23] have been developed, but these provide poor posterior approximations or require first finding a low-cost posterior approximation. The current state-of-the-art approach resolves both of the aforementioned issues by formulating the coreset construction problem as variational inference in the family of coresets [21]:

$$w^\star = \underset{w \in \mathbb{R}^N}{\arg\min} \quad \mathrm{D}_{\mathrm{KL}}\left(\pi_w \| \pi_1\right) \quad \text{s.t.} \quad w \geq 0, \|w\|_0 \leq M, \tag{3}$$

where $\mathrm{D}_{\mathrm{KL}}(\cdot\|\cdot)$ denotes the KL divergence. Noting that coresets are a sparse subset of an exponential family—the weights form the natural parameter $w \in \mathbb{R}^N_{\geq 0}$, the component potentials $(f_n(\theta))_{n=1}^N$ form the sufficient statistic, $\log Z(w)$ is the log partition function, and π_0 is the base density,

$$\pi_w(\theta) \coloneqq \exp\left(w^T f(\theta) - \log Z(w)\right) \pi_0(\theta), \quad f(\theta) \coloneqq (f_1(\theta), \ldots, f_N(\theta))^T \in \mathbb{R}^N, \tag{4}$$

one can obtain a formula for the gradient

$$\nabla_w \mathrm{D}_{\mathrm{KL}}\left(\pi_w \| \pi_1\right) = -\nabla_w^2 \log Z(w)(1-w) = -\operatorname{Cov}_w\left[f, f^T(1-w)\right], \tag{5}$$

where Cov_w denotes covariance under π_w. The method of [21] then involves selecting a data point to add to the coreset, optimizing the coreset weights using stochastic gradient estimates based on Eq. (5), and then iterating. In practice, this method is infeasibly slow outside of small-data problems. For every data point selected it requires sampling from π_w a number of times equal to the number of optimization steps (set to 100 in [21]). This can easily lead to tens of thousands of sampling steps in total, each of which typically necessitates the use of an MCMC algorithm. Conversely, our method requires tens of sampling steps in total.

3 Construction via subsampling and quasi-Newton refinement

In this section, we provide a new Bayesian coreset construction algorithm (Algorithm 1) to solve the sparse variational inference problem Eq. (3). This method involves first uniformly subsampling the data, and then optimizing the weights on the subsample. Here, we give a detailed explanation of how the algorithm works. We defer to Section 4 the theoretical analysis of the method, which demonstrates that, with high probability, it finds a near-optimal coreset in a small number of optimization iterations.

3.1 Uniform subsampling

The first key insight to our approach is that we can select the M data points that comprise the coreset by simple uniform subsampling. From an algorithmic standpoint, the benefit of this approach is clear: it makes the selection of coreset points fast and easy to implement. It also decouples the subset selection from the optimizing of the coreset weights, leaving us with a simpler optimization problem to solve.[1] Intuitively, this is reasonable from a coreset quality standpoint because a uniform subsample of the data will, with high probability, create a highly-expressive "basis" of log-likelihood functions $(f_n)_{n=1}^M$ with which to approximate the full data log-likelihood. Of course, we still need to optimize the coreset weights; but we have not limited the flexibility of the coreset family significantly by selecting points via uniform subsampling. Theorem 4.2 provides a precise statement of these ideas, and shows that the optimal weighted coreset posterior built using a uniformly random subset of data is a good posterior approximation.

Since the coreset data points are chosen uniformly randomly, we will w.l.o.g. re-label the selected subset to have indices $1, \ldots, M$, and the remaining data points to have indices $M+1, \ldots, N$ from now on. We can specify a new weight constraint set $\mathcal{W} \subset \mathbb{R}^N$ of vectors w with 0 entries for every index beyond M, and a further subset $\mathcal{W}_N$ where the weights sum to N,

$$\mathcal{W} = \left\{w \in \mathbb{R}^N : w \geq 0,\ n > M \implies w_n = 0\right\}, \qquad \mathcal{W}_N = \left\{w \in \mathcal{W} : 1^T w = N\right\}.$$

It is also useful to denote $g(\theta) = (f_1(\theta), \ldots, f_M(\theta))^T \in \mathbb{R}^M$. as the first M components of $f(\theta)$.

3.2 Quasi-Newton optimization

Given the choice of M data points to include in the coreset, it is crucial to optimize their weights correctly. This is the second step of our algorithm. The KL divergence gradient for the M weights is

$$\nabla_w \mathrm{D}_{\mathrm{KL}}\left(\pi_w \| \pi_1\right) = \left(-\nabla_w^2 \log Z(w)(1-w)\right)_{1:M} = -\operatorname{Cov}_w\left[g, f^T(1-w)\right]. \tag{6}$$

To reduce the number of weight optimization iterations —each of which generally involves MCMC sampling from π_w, where w is the current set of weights—we develop a second-order optimization method. Again using the properties of exponential families, we can derive the Hessian of the KL divergence with respect to the M weights

$$\nabla_w^2 \mathrm{D}_{\mathrm{KL}}\left(\pi_w \| \pi_1\right) = \left(\nabla_w^2 \log Z(w) - \nabla_w^3 \log Z(w)(1-w)\right)_{1:M,1:M}.$$

This matrix is not guaranteed to be positive (semi-)definite; but near the optimum ,where the coreset approximation is ideally good, i.e., $w^T f(\theta) \approx 1^T f(\theta)$, the first term should dominate the expression. This motivates the use of

$$\nabla_w^2 \mathrm{D}_{\mathrm{KL}}\left(\pi_w \| \pi_1\right) \approx \left(\nabla_w^2 \log Z(w)\right)_{1:M,1:M} = \operatorname{Cov}_w\left[g, g\right] \tag{7}$$

[1] The benefits of this initial uniformly random selection step are also noted by [24, 25] in concurrent work.

to scale gradient steps in the optimization method rather than the true Hessian, thus creating a quasi-Newton method [26]. This heuristic motivates our approach, but it is rigorously justified by Theorem 4.3, which proves that this scaling results in optimization iterations that are guaranteed to converge exponentially to a coreset which is nearly globally optimal. Note that when M is larger than the inherent dimension of the space of log-likelihood functions, $\mathrm{Cov}_w[g, g]$ will have zero eigenvalues and be noninvertible. Therefore we add a regularization $\tau > 0$ prior to inversion. Theorem 4.3 shows how the regularization influences the optimization, but in general we want τ to be as small as possible while still ensuring the numerical stability of inverting $\mathrm{Cov}_w[g, g]$.

Our optimization method is as follows. First, we initialize at the uniformly weighted coreset, i.e.,

$$w_0 \in \mathcal{W}, \quad w_{0m} = \frac{N}{M}, \quad m = 1, \ldots, M.$$

Next, given a step-size tuning sequence γ_k, $k \geq 0$, we update the weights at each iteration $k \in \mathbb{N} \cup \{0\}$ using the $(\tau > 0)$-regularized quasi-Newton step

$$\hat{w}_{k+1} = w_k + \gamma_k (G(w_k) + \tau I)^{-1} H(w_k)(1 - w_k) \tag{8}$$

$$G(w) = \mathrm{Cov}_w[g, g], \quad H(w) = \mathrm{Cov}_w[g, f]. \tag{9}$$

Theorem 4.3 suggests that the step size γ_k should be constant: $\gamma_k = \gamma \in (0, 1]$; but the analysis in that section assumes that we can compute $G(w)$ and $H(w)$ exactly, whereas in practice we will have to estimate them. Hence we allow γ_k to depend on the iteration number in general. Finally, we project $\hat{w}_{k+1}$ back onto the constraint set to obtain next iterate w_{k+1}, i.e. for $m = 1, \ldots, M$ set

$$w_{k+1,m} = \max(\hat{w}_{k+1,m}, 0). \tag{10}$$

3.3 Algorithm

The pseudocode for the quasi-Newton coreset construction method is shown in Algorithm 1. There are a number of practical considerations needed to use this method; we discuss these here.

The first consideration is that we cannot calculate $G(w)$ and $H(w)$ in closed form. Instead, at step k of the algorithm, we use a Monte Carlo estimate; first taking S samples $(\theta_s)_{s=1}^S \overset{\text{i.i.d.}}{\sim} \pi_{w_k}$, and calculating $\hat{g}_s := g(\theta_s) - \frac{1}{S}\sum_{r=1}^S g(\theta_r)$, so that we may estimate $G(w_k)$ as

$$\hat{G}_k = \frac{1}{S}\sum_{s=1}^S \hat{g}_s \hat{g}_s^T \in \mathbb{R}^{M \times M}. \tag{11}$$

As $H(w_k)$ is an $M \times N$ matrix, estimating it directly will incur a $\Theta(MN)$ storage cost. Instead, we note that we only require $H(w_k)(1 - w_k) = \mathrm{Cov}_{w_k}\left[g, f^T(1 - w_k)\right] = \mathrm{Cov}_{w_k}[g, h]$, where $h := f^T 1 - g^T w_k = \bar{f} - g^T w_k$. We can estimate h using $\hat{h}_s = \sum_{n=1}^N \left[f_n(\theta_s) - \frac{1}{S}\sum_{r=1}^S f_n(\theta_r)\right] - \hat{g}_s^T w_k$, which requires $O(N)$ time, but only $O(S)$ space. We can then estimate $H(w_k)(1 - w_k)$ as

$$\hat{H}_k(1 - w_k) = \frac{1}{S}\sum_{s=1}^S \hat{g}_s \hat{h}_s. \tag{12}$$

Thus, we can find w_{k+1} from w_k by taking the stochastic Newton step

$$\hat{w}_{k+1} = w_k + \gamma_k \left(\hat{G}_k + \tau I\right)^{-1} \hat{H}_k(1 - w_k), \tag{13}$$

and projecting onto the constraint set: $w_{k+1,m} = \hat{w}_{k+1,m} \mathbb{1}_{\hat{w}_{k+1,m} \geq 0}$. We set the regularization parameter τ by examining the condition number of $\hat{G}_k + \tau I$ and keeping it below a reasonable value. We can tune γ_k using a line search method. As we do not have access to the objective function (i.e. the KL-divergence between the coreset and full posteriors), we use the curvature part of the Wolfe conditions [27] to tune this. In practice, this line search is expensive. Thus, we only tune γ_k for $k \leq K_{\text{tune}}$, and leave it as a constant thereafter. The intuition here is that we may start with quite poor coreset weights, and so need to choose the initial steps carefully.

Computing $\hat{G}_k + \tau I$ involves taking S samples, forming the product $\hat{g}_s \hat{g}_s^T$, and then inverting the resulting matrix. Its time complexity is thus $O(SM^2 + M^3)$. Computing $\hat{H}(w_k)(1 - w_k)$ similarly has

Algorithm 1 QNC (QUASI-NEWTON CORESET)

Require: $(f_n(\theta))_{n=1}^N$ π_0, S, K, γ, τ, M.
Uniformly sample M data points, w.l.o.g. these correspond to indicies $1, \dots, M$.
Set $w_{0m} = \frac{N}{M}, \quad m = 1, \dots, M$.
for $k = 0, \dots, K-1$ **do**
  Sample $(\theta_s)_{s=1}^S \overset{\text{i.i.d.}}{\sim} \pi_{w_k} \propto \exp\left(w_k^T f(\theta)\right) \pi_0(\theta)$
  Set $\hat{g}_s \leftarrow g(\theta_s) - \frac{1}{S}\sum_{r=1}^S g(\theta_r)$ and $\hat{h}_s \leftarrow \sum_{n=1}^N \left[f_n(\theta_s) - \frac{1}{S}\sum_{r=1}^S f_n(\theta_r)\right] - \hat{g}_s^T w_k$
  Set $\hat{H}_k(1 - w_k) \leftarrow \frac{1}{S}\sum_{s=1}^S \hat{g}_s \hat{h}_s$ and $\hat{G}_k \leftarrow \frac{1}{S}\sum_{s=1}^S \hat{g}_s \hat{g}_s^T$
  if $k \leq K_{\text{tune}}$ **then**
    Choose γ_k via line search with starting value γ
  else
    Set $\gamma_k \leftarrow \gamma$
  Take a quasi-Newton step: $\hat{w}_{k+1} \leftarrow w_k + \gamma_k \left(\hat{G}_k + \tau I\right)^{-1} \hat{H}_k(1 - w_k)$
  Project: for all $m \in [M]$, $w_{k+1,m} \leftarrow \max\left(\hat{w}_{k+1,m}, 0\right)$
return $w = w_K$

complexity $O\left(SM + SN\right)$, and computing the product with $(\hat{G}_k + \tau I)^{-1}$ has complexity $O\left(M^2\right)$. For K Newton steps in Algorithm 1, the overall time complexity is thus $O(K(M^3 + SM^2 + SN))$. The space complexity is $O(M^2)$ (to store $\hat{G}_k$). This is sublinear in N, unlike in [20, 23].

Finally, we can reduce the time complexity by using a further stochastic estimate of $\hat{H}_k(1 - w_k)$. In particular, we can instead calculate $\hat{h}_s$ only for indices in $\mathcal{I} \coloneqq \{1, \dots, M\} \cup \mathcal{T}$, where $\mathcal{T}$ is a uniformly selected sample $\mathcal{T} \subseteq \{1, \dots, N\}$. Calculating $\hat{H}(w_k)(1 - w_k)$ using this subsampled vector $\hat{h}_{s,\mathcal{I}}$ then has complexity $O(SM + ST)$ in the worst case, where $T \coloneqq |\mathcal{T}|$. The overall time complexity is then $O(K(M^3 + SM^2 + ST))$.

4 Theoretical analysis

In this section we provide a theoretical analysis of the proposed method. We demonstrate (in Theorems 4.1 and 4.2) that the optimal coreset posterior built using a uniformly random subset of data is, with high probability, an exact or near-exact approximation of the true posterior. Second, we show (in Theorem 4.3) that the proposed optimization algorithm converges exponentially quickly to a point near the optimal coreset, thus realizing the guarantee in Theorem 4.2 in practice, and theoretically justifying our proposed algorithm. The assumptions required for our theory are somewhat technical. We give some intuition here on settings where they hold, but discuss this further in Appendix A. Proofs of all results may be found in Appendix B.

4.1 Subset selection via uniform subsampling

Theorem 4.1 and Theorem 4.2 provide a theoretical foundation for selection via uniform subsampling. Both recommend setting $M \gtrsim d \log N$, where d is essentially the "dimension" of the span of the log-likelihood functions. Theorem 4.1 states that if the span of the log-likelihood functions is indeed finite-dimensional with dimension d—e.g., in a usual exponential family model—then with high probability, the optimal coreset using a uniformly random selection of M points is exact. Theorem 4.2 extends this to the general setting where $\Theta \subseteq \mathbb{R}^D$, N is large relative to D, and where we assume the posterior concentrates as $N \to \infty$. In particular, Theorem 4.2 states that with high probability, the optimal coreset using a uniformly random selection of M data points provides low error.

In what follows, we assume that $f_n(\cdot)$ are i.i.d. from some true data-generating distribution p, and we define $\bar{f}(\cdot) = \mathbb{E}_p(f_n(\cdot))$. We also introduce the new notation $f(\cdot, \cdot)$, where we consider the potentials f as functions of both x and θ. This enables us to make the data-dependence explicit when needed, i.e., $f_n(\cdot) = f(x_n, \cdot)$; for instance, in the standard context of Bayesian inference $f_n(\cdot) = f(x_n, \cdot) = \log p(x_n|\cdot)$, with $x_n \overset{\text{i.i.d.}}{\sim} p_0$ for $n \in [N]$.

Theorem 4.1. *Let S_0 be the vector space of functions on Θ spanned by $\{f(x,\cdot); x \in \mathcal{X}\}$. Assume that S_0 is finite dimensional, with dimension d. Let μ be any measure on Θ such that $S_0 \subset L^2(\mu)$ and denote $r_\mu^2 = \mathbb{E}_\mu\left(\mathbb{E}_p([f_1 - \bar{f}]^2)\right)$. For $\delta > 0$ define*

$$J(\delta) = \inf_{a \in S_0; \|a\|=1} \Pr\left(\langle a, f_1 - \bar{f}\rangle_{L^2(\mu)} > \frac{2r_\mu}{\sqrt{N\delta}}\right). \tag{14}$$

Then for $\delta > 0$ such that $J(\delta) + \delta < 1$, a universal constant C_1, and $M \in \mathbb{N}$ such that

$$M \geq \frac{2d}{J(\delta)}\left(\log N + \log\left(\frac{4\delta}{J(\delta)}\right) + C_1\right),$$

we have that with probability $\geq 1 - \delta - e^{-MJ(\delta)/4}$,

$$\min_{w \in \mathcal{W}_N} \mathrm{KL}(\pi_w || \pi) = 0.$$

Note that the condition that S_0 has finite dimension $d < \infty$ is satisfied for all d-dimensional exponential families when $f_n(\theta) = \log p(x_n|\theta)$. Further, note that Theorem 4.1 still applies in situations where d increases with N. To understand the behaviour of $J(\delta)$, consider a full rank exponential family with sufficient statistic $B(\cdot)$:

$$(f_1 - \bar{f})(\theta) = A(\theta)^T(B(X_1) - \mathbb{E}_p(B(X))) + K(X_1).$$

If there exists μ such that $\int A(\theta)A(\theta)^T d\mu(\theta)$ has rank d, then

$$J(\delta) \xrightarrow{\delta N \to \infty} \inf_{a \in S_0; \|a\|=1} \Pr\left(\langle a, f_1 - \bar{f}\rangle_{L^2(\mu)} \geq 0\right) > 0.$$

In this case, we can set $\delta = \omega(1/N)$, such that with probability $\geq 1 - \omega(1/N)$, we indeed have that $\min_{w \in \mathcal{W}_N} \mathrm{KL}(\pi_w||\pi) = 0$ for $M \gtrsim d \log N$.

In the interest of obtaining a more general approximation result, we now consider the case where S_0 is infinite dimensional, but where we can find a finite-dimensional space S_1 which approximates $f_n(\theta) - \bar{f}(\theta)$ reasonably well. A typical setup where this is valid is when $\Theta \subseteq \mathbb{R}^D$, the posterior concentrates (as $N \to \infty$) at a point $\theta_0 \in \Theta$ that maximizes $\bar{f}(\theta)$, and each $f_n(\cdot)$ is smooth near θ_0 and globally well-behaved in some sense. Theorem 4.2 below states, roughly, that for a model satisfying these assumptions, $\min_{w \in \mathcal{W}_N} \mathrm{KL}(\pi_w||\pi)$ will be small with high probability. We give the exact conditions in Appendix A. Although they are perhaps strong, the result is still indicative of when a good posterior approximation is obtained by our method.

Intuitively, we require that $f(x,\theta)$ be differentiable, strongly concave in θ, and "smooth enough". Letting θ_0 be the unique value such that $\bar{f}(\theta) \leq \bar{f}(\theta_0)\ \forall \theta \in \Theta$, we also require that we can split $f(x;\theta)$ into two parts,

$$f(x,\theta) = f^{(1)}(x,\theta) + f^{(2)}(x,\theta),$$

where the span of $\left\{f^{(1)}(x,\cdot) - f^{(1)}(x,\theta_0) : x \in \mathcal{X}\right\}$ is a finite-dimensional vector space S_1. This split needs to be made such that $f^{(2)}$ is negligible, in the sense that $f(x,\theta) \approx f^{(1)}(x,\theta)$ locally around θ_0. A usual case where these conditions hold is when $f(x;\theta)$ is C^3 in θ. In this case, we can obtain $f^{(1)}(x;\theta)$ by a second-order Taylor expansion,

$$f^{(1)}(x;\theta) = f(x;\theta_0) + (\theta - \theta_0)^T\nabla_\theta f(x;\theta_0) + (\theta - \theta_0)^T\nabla^2 f(x;\theta_0)(\theta - \theta_0)/2,$$

which has dimension bounded by $1 + D + D(D+1)/2$. $f^{(2)}(x;\theta)$ is then the remainder term, which is indeed negligible around θ_0 in the desired sense. In Appendix A, we provide further discussion on situations when the necessary conditions for Theorem 4.2 hold. Recalling that a quantity is $o(1)$ as $N \to \infty$ if and only if it converges to 0, we can then state the result as follows.

Theorem 4.2. *Suppose the assumptions A1-A4 in Appendix A hold, and we set $M \gtrsim D(\log N + 1)$. Then as $N \to \infty$, with probability at least $1 - o(1)$, we have that*

$$\min_{w \in \mathcal{W}_N} \mathrm{KL}(\pi_w||\pi) = o(1).$$

In the above example, we can examine the proof to obtain a bound on $\mathrm{KL}(\pi_w||\pi)$ of order $1/\sqrt{N}$ with probability converging to 1, showing that this result does indeed lead to a useful bound.

4.2 Weight optimization via approximate Newton

Theorems 4.1 and 4.2 guarantee the existence of a high-quality coreset, but do not provide any insight into whether it is possible to find it tractably. Theorem 4.3 addresses this remaining gap. In particular, intuitively, it asserts that as long as (1) the regularization parameter $\tau > 0$ is nonzero but small enough that it does not interfere with the optimization (in particular, the minimum eigenvalue of $G(w)$ over the space), and (2) the *optimal* coreset is a good approximation to the full dataset (which is already guaranteed by Theorem 4.2), then the weights w_k at iteration k of the approximate Newton method converge exponentially to a point close to a global optimum. In the result below, let $G(w)$ and $H(w)$ be defined as in Eq. (9).

Theorem 4.3. *Let $W \subseteq \mathcal{W}$ be a closed convex set, $W^\star \subseteq \operatorname{argmin}_{w \in W} \mathrm{KL}(\pi_w || \pi)$ be a maximal closed convex subset, and fix the regularization parameter $\tau > 0$. Define*

$$\xi = \inf_{w \in W} \min \left\{ \frac{\lambda}{\lambda + \tau} : \lambda \in \operatorname{eigvals} G(w),\, \lambda > 0 \right\}.$$

Suppose $\exists\, \epsilon \in [0, 1)$ and $\delta \geq 0$ such that for all $w \in W$

$$\left\| \left(G(w) + \tau I\right)^{-1} H(w)(1 - w^\star) \right\| \leq \epsilon \|w - w^\star\| + \delta, \tag{15}$$

where $w^\star = \operatorname{proj}_{W^\star}(w)$. Then the k^{th} iterate $w_k \in W$ and its projection $w_k^\star = \operatorname{proj}_{W^\star}(w_k)$ of the projected Newton method defined in Section 3.2 by Eqs. (8)*,* (9) *and* (10)*, initialized at $w_0 \in W$ with a fixed step size $\gamma \in [0, 1]$ and regularization τ satisfies*

$$\|w_k - w_k^\star\| \leq \eta^k \|w_0 - w_0^\star\| + \gamma\delta \left(\frac{1 - \eta^k}{1 - \eta} \right), \qquad \eta = 1 - \gamma(\xi - \epsilon).$$

In order to use this result, the minimum positive eigenvalue of $G(w)$ for $w \in W \subseteq \mathcal{W}$ needs to be bounded, as does the error of the optimal coreset weights $w^\star$ in the sense of Eq. (15). A notable corollary of Theorem 4.3 occurs when there exists a $w^\star$ such that

$$\forall \theta \in \Theta, \quad \sum_{m=1}^{M} w_m^\star f_m(\theta) = \sum_n f_n(\theta),$$

i.e., the optimal coreset is identical to the true posterior distribution, as is guaranteed by Theorem 4.1. For any model where the conditions of Theorem 4.1 hold, such as any exponential family model, Eq. (15) holds with $\epsilon = \delta = 0$. We can thus take $\gamma = 1$ to find that

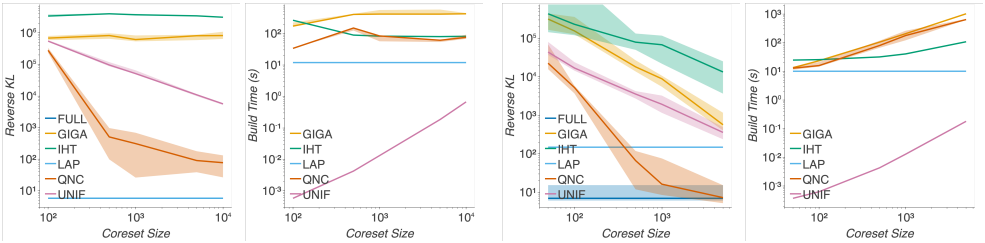
$$\|w_k - w_k^\star\| \leq (1 - \xi)^k \|w_0 - w_0^\star\|.$$

This provides some intuition on Theorem 4.3; as long as the optimal coreset posterior is a reasonable approximation to π, and we use a small regularization $\tau > 0$ (such that $\xi \approx 1$), then we generally need only a small number of optimization steps to find the optimal coreset. The conditions of Theorem 4.1 are sufficient, but not necessary, for Eq. (15) to hold. We present further discussion in Appendix A.

5 Experiments

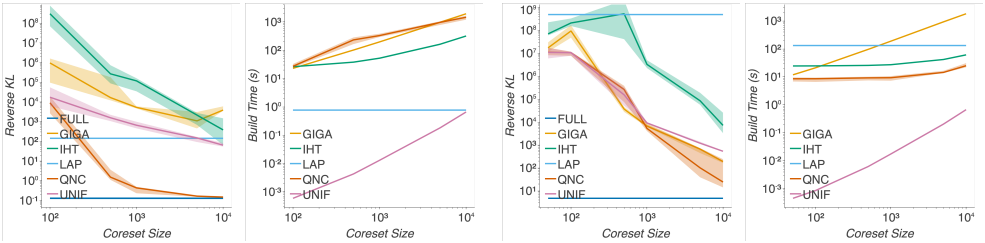
In this section, we compare our proposed quasi-Newton coreset (QNC) construction against existing constructions—uniform subsampling (UNIF), greedy iterative geodesic ascent (GIGA) [20] and iterative hard thresholding (IHT) [23]—as well as the Laplace approximation (LAP), which represents what one obtains by assuming posterior normality in the large-data setting. In each case, we use $S = 500$ Monte Carlo samples during coreset construction. We compute error metrics using 1000 samples from each method's approximate posterior. We also compare to the baseline of sampling from the full posterior (FULL) to establish a noise floor for the given comparison sample size of 1000. In the synthetic Gaussian experiment, which is the simplest one we consider, we see that sparse variational inference (SVI) [21] is prohibitively slow, taking 4.2×10^3s to construct a coreset of size 1 (and scaling at best linearly with coreset size). Thus, we do not compare against it.

For GIGA and IHT, we need to supply a low-cost approximation $\hat{\pi}$ to the posterior. To ensure these methods apply as generally as QNC and SVI, we use a uniformly sampled coreset approximation of size M with weights N/M (where M is the same as the desired coreset size we are constructing).



(a) Gaussian

(b) Sparse linear regression



(c) Logistic regression

(d) Basis function regression

Figure 1: Reverse KL divergence (left) and build time in seconds (right) for each experiment. We plot the median and a shaded area between the 25$^{\text{th}}$/75$^{\text{th}}$ percentiles over 10 random trials. Our algorithm (QNC) provides an improvement in coreset quality, with a comparable run-time and less user input.

We also use these weights for UNIF. In Appendix C we provide additional results for GIGA and IHT with a Laplace approximation used for $\hat{\pi}$.

Experiments are performed in settings standard in the literature, but with larger dataset sizes than those that could be considered previously. For each experiment we plot the approximate reverse KL divergence obtained by assuming posterior normality, and the build time. In Appendix C we provide comparisons on the additional metrics of relative mean and covariance error, forward KL divergence and per sample time to sample from the respective posteriors. For the experiments in Sections 5.2 and 5.3 with heavy tailed priors, we also compare using the maximum mean discrepancy [28] and kernel Stein discrepancy [29, 30], with inverse multi-quadratic (IMQ) kernel. Experiments were performed on a machine with a 2.6GHz 6-Core Intel Core i7 processor, and 16GB memory.

Throughout the experiments, we see that QNC outperforms the other subsampling methods for most coreset sizes we consider. GIGA and IHT in particular are limited by a fixed choice of $\hat{\pi}$ and finite projection dimension defined by our choice of $S = 500$—we discuss this further in Appendix C. Furthermore, outside of the synthetic Gaussian experiment, we see that QNC outperforms LAP for coreset sizes above certain thresholds, which represent a small fraction of the full dataset.

5.1 Synthetic Gaussian location model

Our first comparison is on a Gaussian location model, with prior $\theta \sim \mathcal{N}(\mu_0, \sigma_0 I)$, and likelihood $(X_n)_{n=1}^N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\theta, \sigma I)$ in D dimensions. Here, we take $\mu_0 = 0$ and $\sigma_0 = 1$. Closed form expressions are available for the subsampled posterior distributions [21, Appendix B], and we can sample from them without Monte Carlo. We compare the methods on a synthetic dataset with $N = 1,000,000$, $D = 100$, where we generate the $(X_n)_{n=1}^N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \sigma I)$ and set $(\mu_i)_{i=1}^D \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 100)$ and $\sigma = 100$. From Fig. 1a we see that our method outperforms the other subsampling methods for all coreset sizes. This example is largely illustrative; we expect LAP to provide the exact posterior here by design, and this is indeed what we find (the reverse KL plots for LAP and FULL overlap).

5.2 Bayesian sparse linear regression

Next, we study a Bayesian sparse linear regression problem, where the data $(x_n, y_n)_{n=1}^N$ consists of a feature $x_n \in \mathbb{R}^D$ and an outcome $y_n \in \mathbb{R}^N$. The posterior distribution is that of $\theta \in \mathbb{R}^D$ in the model

$$y_n = x_n^T \theta + \epsilon_n, \quad n = 1, \ldots, N \tag{16}$$

where $\epsilon_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma),\ n = 1, \ldots, N$. We place independent Cauchy priors on the coefficients θ:

$$\theta_i \overset{\text{i.i.d.}}{\sim} \text{Cauchy}(0, \tau\lambda_i), \quad i = 1, \ldots, D,$$

where the hyperpriors are $\lambda_i \overset{\text{i.i.d.}}{\sim} \text{Half-Cauchy}(0, 1)$, $\tau \sim \text{Half-Cauchy}(0, \sigma_0)$, with $\sigma_0 = 2$, and $\sigma \sim \Gamma(a_0, b_0)$ with $a_0 = 1, b_0 = 1$. We perform posterior inference on the $2D + 2-$dimensional set of parameters $(\theta_1, \ldots, \theta_D, \lambda_1, \ldots, \lambda_D, \sigma, \tau)$. Sampling is performed using STAN [31]. The dataset we study is a flight delays dataset,[2] with $N = 100,000$ and $D = 13$ (so that the overall inference problem is $28-$dimensional). The response variable y_n is the delay in the departure time of a flight, and the features are meteorological and flight-specific information. We see in Fig. 1b that our method outperforms the other subsampling methods for all coreset sizes, and LAP for sizes above 500—representing 0.5% of the data. In Appendix C we see that the target posterior in this case has heavy tails, which makes the Laplace approximation particularly unsuited to this problem. We can see this even more clearly in the additional results presented in Appendix C.

5.3 Heavy-tailed Bayesian logistic regression

For this comparison, we perform Bayesian logistic regression with parameter $\theta \in \mathbb{R}^D$ having a heavy-tailed Cauchy prior $\theta_i \overset{\text{i.i.d.}}{\sim} \text{Cauchy}(0, \sigma)$, $i = 1, \ldots, D$. The data $(x_n, y_n)_{n=1}^N$ consists of a feature $x_n \in \mathbb{R}^D$ and a label $y_n \in \{-1, 1\}$. The relevant posterior distribution is that of $\theta \in \mathbb{R}^D$ which governs the generation of y_n given x_n via

$$y_n \mid x_n, \theta \overset{\text{indep}}{\sim} \text{Bern}\left(\frac{1}{1 + e^{-x_n^T\theta}}\right). \tag{17}$$

Again, sampling is performed using STAN. As in Section 5.2, we use the flight delays dataset, so we have that $N = 100,000$ and $D = 13$. The response variables y_n are binarized, so that $y_n = 1$ if a flight was cancelled or delayed by more than an hour, and $y_n = -1$ if it was not. In Fig. 1c we see that our method outperforms other subsampling methods for all coreset sizes, and LAP for sizes above 500—representing 0.5% of the data.

5.4 Bayesian radial basis function regression

Our final comparison is on a Bayesian basis function regression example. This is a larger version of the same experiment performed by [21], with $D = 301$ as before, but $N = 100,000$. Full details are given in Appendix C. From Fig. 1d we see that our method provides a significant improvement over LAP for all coreset sizes, which performs particularly poorly in this experiment, despite the Gaussianity of the target posterior [21, Appendix B]. It also outperforms the other subsampling methods for coreset sizes of 1000 and above (representing 1% of the data) but this difference is not as marked as in the other examples. The fact that QNC requires a large coreset to start providing a benefit beyond UNIF stems from the fact that, in this higher dimensional example, there are sometimes data points that are very influential for a certain basis, and both UNIF and QNC can miss these. This indicates the need (in some settings) for a secondary method that can check for and include these, without the significant cost that e.g. SVI entails. We defer this study to future work. However, we do note that for this example QNC has the lowest build time, even outperforming LAP. This is because we only perform a small number of quasi-Newton steps before no further improvement is possible.

6 Conclusion

This paper introduces a novel method for data summarization prior to Bayesian inference. In particular, the method first selects a small subset of data uniformly randomly, and then optimizes the weights on those data points using a novel quasi-Newton method. Theoretical results demonstrate that the method is guaranteed to find a near-optimal coreset, and that the optimal coreset has a low KL divergence to the posterior with high probability. Future work includes investigating the performance of the method in more complex models, and studying conditions under which the method can provide compression in time sublinear in dataset size.

[2] This dataset was constructed by merging airport on-time data from the US Bureau of Transportation Statistics `https://www.transtats.bts.gov/DL_SelectFields.asp?gnoyr_VQ=FGJ` with historical weather records from `https://wunderground.com`.

Acknowledgments and Disclosure of Funding

T. Campbell was supported by a National Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant and Discovery Launch Supplement. C. Naik was supported by the Engineering and Physical Sciences Research Council and Medical Research Council [award reference 1930478]. J. Rousseau was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 834175).

References

[1] Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.

[2] Christian Robert and George Casella. A short history of Markov Chain Monte Carlo: subjective recollections from incomplete data. *Statistical Science*, 26(1):102–115, 2011.

[3] Andrew Gelman, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin. *Bayesian data analysis*. CRC Press, 3rd edition, 2013.

[4] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[5] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *International Conference on Artificial Intelligence and Statistics*, 2014.

[6] Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On Markov chain Monte Carlo methods for tall data. *The Journal of Machine Learning Research*, 18(1):1515–1557, 2017.

[7] Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in MCMC land: cutting the Metropolis-Hastings budget. In *International Conference on Machine Learning*, 2014.

[8] Dougal Maclaurin and Ryan Adams. Firefly Monte Carlo: exact MCMC with subsets of data. In *Conference on Uncertainty in Artificial Intelligence*, 2014.

[9] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, 2011.

[10] Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *International Conference on Machine Learning*, 2012.

[11] Matias Quiroz, Robert Kohn, and Khue-Dung Dang. Subsampling MCMC—an introduction for the survey statistician. *Sankhya: The Indian Journal of Statistics*, 80-A:S33–S69, 2018.

[12] Joris Bierkens, Paul Fearnhead, and Gareth Roberts. The zig-zag process and super-efficient sampling for bayesian analysis of big data. *The Annals of Statistics*, 47(3):1288–1320, 2019.

[13] Murray Pollock, Paul Fearnhead, Adam M Johansen, and Gareth O Roberts. Quasi-stationary monte carlo and the scale algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(5):1167–1221, 2020.

[14] James Johndrow, Natesh Pillai, and Aaron Smith. No free lunch for approximate MCMC. *arXiv:2010.12514*, 2020.

[15] Tigran Nagapetyan, Andrew Duncan, Leonard Hasenclever, Sebastian Vollmer, Lukasz Szpruch, and Konstantinos Zygalakis. The true cost of stochastic gradient Langevin dynamics. *arXiv:1706.02692*, 2017.

[16] Michael Betancourt. The fundamental incompatibility of Hamiltonian Monte Carlo and data subsampling. In *International Conference on Machine Learning*, 2015.

[17] Matias Quiroz, Robert Kohn, Mattias Villani, and Minh-Ngoc Tran. Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*, 114(526):831–843, 2019.

[18] Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable Bayesian logistic regression. In *Advances in Neural Information Processing Systems*, 2016.

[19] Trevor Campbell and Tamara Broderick. Automated scalable Bayesian inference via Hilbert coresets. *Journal of Machine Learning Research*, 20(15), 2019.

[20] Trevor Campbell and Tamara Broderick. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, 2018.

[21] Trevor Campbell and Boyan Beronov. Sparse variational inference: Bayesian coresets from scratch. In *Advances in Neural Information Processing Systems*, 2019.

[22] Dionysis Manousakas, Zuheng Xu, Cecilia Mascolo, and Trevor Campbell. Bayesian pseudo-coresets. In *Advances in Neural Information Processing Systems*, 2020.

[23] Jacky Zhang, Rajiv Khanna, Anastasios Kyrillidis, and Oluwasanmi Koyejo. Bayesian coresets: revisiting the nonconvex optimization perspective. In *Artificial Intelligence in Statistics*, 2021.

[24] Martin Jankowiak and Du Phan. Surrogate likelihoods for variational annealed importance sampling. *arXiv preprint arXiv:2112.12194*, 2021.

[25] Naitong Chen, Zuheng Xu, and Trevor Campbell. Bayesian inference via sparse hamiltonian flows. *arXiv preprint arXiv:2203.05723*, 2022.

[26] Charles Broyden. Quasi-Newton methods. In W Murray, editor, *Numerical Methods for Unconstrained Optimization*, pages 87–106. Academic Press, 1972.

[27] Philip Wolfe. Convergence conditions for ascent methods. *SIAM review*, 11(2):226–235, 1969.

[28] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[29] Qiang Liu, Jason Lee, and Michael Jordan. A kernelized Stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284. PMLR, 2016.

[30] Anna Korba, Pierre-Cyril Aubin-Frankowski, Szymon Majewski, and Pierre Ablin. Kernel Stein discrepancy descent. *arXiv preprint arXiv:2105.09994*, 2021.

[31] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1):1–32, 2017.

[32] Károly Böröczky and Gergely Wintsche. Covering the sphere by equal spherical balls. In Boris Aronov, Saugata Basu, János Pach, and Micha Sharir, editors, *Discrete and Computational Geometry*, volume 25 of *Algorithms and Combinatorics*, pages 235–251. Springer, 2003.

[33] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

A Theoretical analysis

In this section we give further details on the theoretical results presented in Section 4. We start by giving the exact conditions needed for Theorem 4.2 to hold.

A.1 Full conditions for Theorem 4.2

In order to state Theorem 4.2, we define θ_0 as the unique value such that $\bar{f}(\theta) \leq \bar{f}(\theta_0)\ \forall \theta \in \Theta$, where $\bar{f}(\cdot) = \mathbb{E}_p(f_n(\cdot))$. If we split $f(x;\theta)$ into two parts,

$$f(x,\theta) = f^{(1)}(x,\theta) + f^{(2)}(x,\theta),$$

then we require that there exist constants $\epsilon_0, L_0, L_1, L_2 > 0$ such that:

- **A1**: For all x, $f(x,\theta)$ is differentiable and strongly concave in θ with constant L_0.
- **A2**: For all $0 < \epsilon < \epsilon_0$ and $\|\theta - \theta_0\| \leq \epsilon$,

$$|f^{(2)}(x,\theta)| \leq R(x) r(\epsilon) \|\theta - \theta_0\|^2,$$

where $R(x), r(\epsilon)$ are positive functions, r is monotone increasing, $\lim_{\epsilon \to 0} r(\epsilon) = 0$, and

$$4 r(\epsilon_0) \left[\mathbb{E}_p(R(X)) + 1\right] \leq L_0, \quad \mathbb{E}_p(R(X)^2) < \infty.$$

- **A3**: For all $\|\theta - \theta_0\| \leq \epsilon_0$,

$$\bar{f}(\theta_0) - \bar{f}(\theta) \leq L_1 \|\theta - \theta_0\|^2, \quad \mathrm{Var}_p(f(X;\theta_0) - f(X;\theta)) \leq L_2 \|\theta - \theta_0\|^2.$$

- **A4**: $\inf_{\|\theta - \theta_0\| \leq \epsilon_0} \pi_0(\theta) > 0$, and $\sup_\theta \pi_0(\theta) < \infty$.

Here, the functions $R(x), r(\epsilon)$ control the size of the "discarded part" $f^{(2)}$ locally around θ_0; these should be small to ensure that $f(x,\theta) \approx f^{(1)}(x,\theta)$ locally. Note also that when $r(\epsilon) \asymp \epsilon$ as $\epsilon \to 0$, the result of Theorem 4.2 holds even if D is increasing such that $D = o(\log N)$.

A.2 Discussion on conditions of Theorems 4.1 to 4.3

In order for Theorem 4.1 to hold, we require that S_0 is finite dimensional (say with dimension d), where S_0 is the vector space of functions on Θ spanned by $\{f(x,\cdot); x \in \mathcal{X}\}$. For this to hold, we need to find a set of d basis functions for the space, even though $\mathcal{X}$ may be potentially uncountable. There are two settings in which we can clearly see that this holds. The first of these is when $f(x_n, \theta) = \log p(x_n|\theta)$ and we can write the likelihood as a d-dimensional exponential family. The standard exponential family form gives us the desired basis for S_0. The second setting is when $\mathcal{X}$ is in fact a finite set. If $x \in \mathcal{X}$ can only take d values, then we can easily find a basis as before.

In order for Theorem 4.2 to hold, we require additional smoothness conditions on $f(x, \theta)$. These conditions are satisfied if, for example, $f(x;\theta)$ is C^3 in θ, as discussed in Section 4. For a concrete example in which this theorem can be applied, we can consider a logistic regression problem. Here, we are not in the exponential family setting. However, the logistic regression log-likelihood is concave and C^3 in θ. Hence we can see that conditions A1-A3 are satisfied, with $f^{(1)}$ and $f^{(2)}$ found via a second-order Taylor expansion. Finally, A4 is satisfied by any prior that is strictly positive and finite, such as the Cauchy prior we use in our logistic regression experiment.

In the statement of Theorem 4.3, we suppose that $\exists\, \epsilon \in [0, 1)$ and $\delta \geq 0$ such that for all $w \in W$

$$\| \left(G(w)+\tau I\right)^{-1} H(w)(1-w^\star)\| \leq \epsilon \|w-w^\star\|+\delta,, \tag{18}$$

where $w^\star = \text{proj}_{W^\star}(w)$ and $W^\star \subseteq \text{argmin}_{w\in W} \text{KL}(\pi_w||\pi)$.

If we can show that

$$\forall \theta \in \Theta, \quad \sum_{m=1}^{M} w_m^\star f_m(\theta) = \sum_n f_n(\theta), \tag{19}$$

then

$$\begin{aligned} H(w)(1-w^\star) &= \text{Cov}_w\left[g, f^T(1-w^\star)\right] \\ &= 0. \end{aligned} \tag{20}$$

and so thus the assumption will hold with $\epsilon = \delta = 0$.

Theorem 4.1 tells us that $\pi_{w^\star}(\theta) = \pi(\theta)$ for almost all values of θ. This implies that Eq. (19) also holds for almost all values of θ, and so Eq. (20) holds. This means that the assumption in Theorem 4.3 does in fact hold with $\epsilon = \delta = 0$.

Thus, in any setting where the conditions of Theorem 4.1 hold (such as the two discussed above), Theorem 4.3 holds as well. This reasoning also gives us some intuition into how we can generalise this. Intuitively, we require that

$$\forall \theta \in \Theta, \quad \sum_{m=1}^{M} w_m^\star f_m(\theta) \approx \sum_n f_n(\theta), \tag{21}$$

so that $H(w)(1-w^\star) \approx 0$. Essentially, we see that the better the quality of the optimal coreset $w^\star$, the tighter the bound we can find for Eq. (18). We conjecture that if $\exists\, \xi$ such that we can uniformly bound

$$\sup_{\theta\in\Theta} \left| \sum_{m=1}^{M} w_m^\star f_m(\theta) - \sum_n f_n(\theta) \right| \leq \xi,$$

then we can find ϵ and δ such that Eq. (18) holds. We leave the theoretical examination of this claim for future work.

B Proofs

In this section we give the full proofs of Theorems 4.1 to 4.3.

B.1 Proof of Theorem 4.1

Proof. We select M indices uniformly at random from $[N]$, corresponding to M potential functions $f_n(\theta)$. Because the functions f_n are generated i.i.d. and the indices chosen uniformly, the particular indices selected are unimportant for this proof; without loss of generality, we can assume the M selected indices are $n = 1, \ldots, M$.

Let μ be a probability measure such that S_0 is the associated d dimensional subspace of $L^2(\mu)$ and let $M \in [N]$. In what follows, we assume that all norms and inner products are the weighted $L^2(\mu)$ norms and inner products respectively, unless otherwise stated. Further, define $g(\cdot) = \sum_{n=1}^{N}[f_n - \bar{f}](\cdot)/N$, noting that this is an abuse of notation from Section 3. We show below that with probability $\geq 1 - \delta$, there exists $v_1, \cdots, v_M \geq 0$ with $\sum_m v_m = 1$ such that

$$\sum_{m=1}^{M}(f_m - \bar{f})v_m = g. \tag{22}$$

For such a $v = (v_1, \cdots, v_M)$, set $w(v) = Nv$ (with the implicit convention that $w(v)_j = 0$ for $j > M$). We note that

$$\begin{aligned} g - \sum_m v_m(f_m - \bar{f}) &= \frac{1}{N}\sum_{n=1}^{N}(f_n - \bar{f}) - \frac{1}{N}\sum_{n=1}^{N} w_n(f_n - \bar{f}) \\ &= \frac{1}{N}\sum_{n=1}^{N}(1 - w_n)f_n, \end{aligned}$$

and thus

$$\begin{aligned} \sum_n f_n - N(1-t)[g - \sum_m v_m(f_m - \bar{f})] &= \sum_n f_n - (1-t)\sum_n (1 - w_n)f_n \\ &= t\sum_n f_n + (1-t)\sum_n w_n f_n. \end{aligned}$$

Hence, denoting

$$\mathbb{E}_{w(v),t}(h) = \frac{\int_\Theta h(\theta)e^{\sum_n f_n(\theta) - N(1-t)[g - \sum_m v_m(f_m - \bar{f})]}\pi_0(\theta)d\theta}{\int_\Theta e^{\sum_n f_n(\theta) - N(1-t)[g - \sum_m v_m(f_m - \bar{f})]}\pi_0(\theta)d\theta},$$

we have that

$$\mathrm{KL}(\pi_{w(v)}||\pi) = N^2\int_0^1 (1-t)\mathbb{E}_{w(v),t}\left[g - \sum_m v_m(f_m - \bar{f}) - \mathbb{E}_{w(v),t}\left(g - \sum_m v_m(f_m - \bar{f})\right)\right]^2 \mathrm{d}t. \tag{23}$$

This expression for the KL divergence is derived in the proof of [21, Proposition 3]. If we can prove Eq. (22), Eq. (23) would give us that $\mathrm{KL}(\pi_{w(v)}||\pi) = 0$, thus completing the proof. Of course, if Eq. (22) holds then the coreset and full posteriors are equal, so $\mathrm{KL}(\pi_{w(v)}||\pi)$ is trivially zero. However, targeting this expression will be helpful for the proof of 4.2. In order to prove Eq. (22), first note that $\|g\|^2_{L^2(\mu)} = \mathbb{E}_\mu\left(\left(\frac{1}{N}\sum_n(f_n - \bar{f})\right)^2\right)$. With $r_\mu^2 = \mathbb{E}_\mu\left(\mathbb{E}_p([f_1 - \bar{f}]^2)\right)$, we have by Markov's inequality that, $\forall \delta > 0$,

$$\begin{aligned} \Pr\left(\|g\|^2_{L^2(\mu)} > \frac{r_\mu^2}{N\delta}\right) &\leq \frac{N\delta}{r_\mu^2}\mathbb{E}_p\left(\mathbb{E}_\mu\left(\left(\frac{1}{N}\sum_n(f_n - \bar{f})\right)^2\right)\right) \\ &= \frac{N\delta}{r_\mu^2}\mathbb{E}_\mu\left(\mathbb{E}_p\left(\left(\frac{1}{N}\sum_n(f_n - \bar{f})\right)^2\right)\right) \\ &= \frac{N\delta}{r_\mu^2}\frac{1}{N}\mathbb{E}_\mu\left(\mathbb{E}_p\left(\left(f_1 - \bar{f}\right)^2\right)\right) \\ &= \delta. \end{aligned}$$

Defining $t_N \coloneqq \sqrt{r_\mu^2/(N\delta)}$, it is enough to show that, with high probability, the convex hull of $(f_m - \bar{f}), m \in [M]$ contains the ball (in S_0) centered at 0 and with radius t_N. As in the theorem statement, S_0 is defined as the vector space of functions on Θ spanned by $\{f(x,\cdot); x \in \mathcal{X}\}$. This boils down to bounding

$$\Pr\left(\inf_{a\in S_0;\|a\|=1} \max_m \langle a, f_m - \bar{f}\rangle \geq t_N\right)$$

from below, since on this event any point in the ball will be in the convex hull. Moreover, with probability $\geq 1-\delta$, g is in this ball, and thus Eq. (22) will be satisfied.

From Böröczky and Wintsche [32, Corollary 1.2], the unit sphere in d-dimensions can be covered by

$$N_d(\phi) = \frac{C\cdot\cos\phi}{\sin^d\phi} d^{\frac{3}{2}}\log(1+d\cos^2\phi) \leq \phi^{-d}A_d, \quad A_d = Ce^{\frac{d}{2}}d^{\frac{3}{2}}\log(1+d)$$

balls of radius $0 < \phi \leq \arccos\frac{1}{\sqrt{d+1}}$ and centers $(a_i)_{i\leq N_d(\phi)}$, where C is a universal constant. Moreover,

$$\inf_{a\in S_0;\|a\|=1}\max_m \langle a, f_m - \bar{f}\rangle \geq \min_{i\leq N_d(\phi)} \max_{m=1,\ldots,M} \left[\langle a_i, f_m - \bar{f}\rangle - \phi\|f_m - \bar{f}\|_{L^2(\mu)}\right].$$

For any $a \in S_0$ with $\|a\| = 1$, by independence, and the union bound,

$$\begin{aligned}
&\Pr\left(\max_{m=1,\ldots,M}\left[\langle a, f_m - \bar{f}\rangle - \phi\|f_m - \bar{f}\|_{L^2(\mu)}\right] \leq t_N\right)\\
&\quad= \Pr\left(\langle a, f_1 - \bar{f}\rangle - \phi\|f_1 - \bar{f}\|_{L^2(\mu)} \leq t_N\right)^M\\
&\quad\leq \left[\Pr\left(\langle a, f_1 - \bar{f}\rangle \leq 2t_N\right) + \Pr\left(\|f_1 - \bar{f}\| > t_N/\phi\right)\right]^M
\end{aligned}$$

We have, for all $a \in S_0$ with $\|a\| = 1$,

$$\begin{aligned}
\Pr(\langle a, f_1 - \bar{f}\rangle > 2t_N) &\geq \inf_{a\in S_0;\|a\|=1}\Pr\left(\langle a, f_1 - \bar{f}\rangle > 2t_N\right)\\
&= J(\delta),
\end{aligned}$$

using the definition of $J(\delta)$ as given in the statement of Theorem 4.1. Thus,

$$\Pr(\langle a, f_1 - \bar{f}\rangle \leq 2t_N) \leq 1 - J(\delta).$$

Furthermore, choosing $\phi^2 = J(\delta)/(4N\delta)$, we have by Chebychev's inequality that

$$\Pr\left(\|f_1 - \bar{f}\| > t_N/\phi\right) \leq J(\delta)/2.$$

Finally we obtain, using the union bound and the above results, that

$$\begin{aligned}
\Pr\left(\inf_{a\in S_0;\|a\|=1}\max_m \langle a, f_m - \bar{f}\rangle < t_N\right) &\leq \Pr\left(\min_{i\leq N_d(\phi)}\max_{m=1,\ldots,M}\left[\langle a_i, f_m - \bar{f}\rangle - \phi\|f_m - \bar{f}\|_{L^2(\mu)}\right] < t_N\right)\\
&\leq \sum_{i\leq N_d(\phi)} \Pr\left(\max_{m=1,\ldots,M}\left[\langle a_i, f_m - \bar{f}\rangle - \phi\|f_m - \bar{f}\|_{L^2(\mu)}\right] < t_N\right)\\
&\leq \sum_{i\leq N_d(\phi)} \left[\Pr\left(\langle a_i, f_1 - \bar{f}\rangle \leq 2t_N\right) + \Pr\left(\|f_1 - \bar{f}\| > t_N/\phi\right)\right]^M\\
&\leq \sum_{i\leq N_d(\phi)} (1 - J(\delta) + J(\delta)/2)^M\\
&= N_d(\phi)(1 - J(\delta)/2)^M.
\end{aligned}$$

For M such that $MJ(\delta) \geq 4\log(N_d(\phi))$, we then have that

$$\begin{aligned}
N_d(\phi)(1 - J(\delta)/2)^M &\leq e^{MJ(\delta)/4}e^{M\log(1-J(\delta)/2)}\\
&\leq e^{MJ(\delta)/4}e^{-MJ(\delta)/2} = e^{-MJ(\delta)/4}.
\end{aligned}$$

Using the union bound, we can therefore see that Eq. (22) holds with probability $\geq 1-\delta-e^{-MJ(\delta)/4}$. Noting that $N_d(\phi) \leq \phi^{-d} A_d$, a sufficient condition on M for this to hold is that

$$\begin{aligned} M &\geq \frac{4}{J(\delta)} \log\left(\phi^{-d} A_d\right) \\ &= \frac{4}{J(\delta)} \log\left(\left(\frac{J(\delta)}{4N\delta}\right)^{-d/2} A_d\right) \\ &= \frac{2d}{J(\delta)}\left[\log N + \log\left(\frac{4\delta}{J(\delta)}\right) + \log\left(A_d^{-d/2}\right)\right] \\ &\geq \frac{2d}{J(\delta)}\left[\log N + \log\left(\frac{4\delta}{J(\delta)}\right) + C_1\right], \end{aligned}$$

where C_1 is a constant such that $C_1 \leq \log\left(A_d^{-2/d}\right)$. This arrives at the condition on M in the theorem statement, and thus completes the proof. □

B.2 Proof of Theorem 4.2

Proof. By the concavity and differentiability of $f(x,\cdot)$ we have $f(x;\theta) \leq f(x;\theta_0) + \nabla_\theta f(x;\theta_0)^T(\theta-\theta_0) - L_0\|\theta-\theta_0\|^2$. Set $S_N = \sum_n \nabla_\theta f_n(\theta_0)/\sqrt{N}$ and recall that by definition of θ_0 as the maximizer of $\bar{f} = \mathbb{E}_p(f_1(\theta))$, $\mathbb{E}_p\left(\nabla_\theta f_1(\theta_0)\right) = 0$.

Starting from Eq. (23) we decompose, for each $t \in (0,1)$, the integral over Θ in the numerator of Eq. (23) into an integral over $B_\epsilon = \{\|\theta-\theta_0\| \leq \epsilon\}$ and an integral over B_ϵ^c. Here, $0 < \epsilon < \epsilon_0$, where ϵ_0 is as defined in the assumptions before the statement of Theorem 4.2. We can rewrite $\mathbb{E}_{w(v),t}(h)$ as

$$\begin{aligned} \mathbb{E}_{w(v),t}(h) &= \frac{\int_\Theta h(\theta) e^{\sum_n f_n(\theta) - N(1-t)[g(\theta) - \sum_m v_m(f_m(\theta)-\bar{f}(\theta))]}\pi_0(\theta)d\theta}{\int_\Theta e^{\sum_n f_n(\theta) - N(1-t)[g(\theta) - \sum_m v_m(f_m(\theta)-\bar{f}(\theta))]}\pi_0(\theta)d\theta} \\ &= \frac{\int_\Theta h(\theta) e^{t\sum_n f_n(\theta) + N(1-t)\sum_m v_m f_m(\theta)}\pi_0(\theta)d\theta}{\int_\Theta e^{t\sum_n f_n(\theta) + N(1-t)\sum_m v_m f_m(\theta)}\pi_0(\theta)d\theta} \times \frac{e^{-t\sum_n f_n(\theta_0) - N(1-t)\sum_m v_m f_m(\theta_0)}}{e^{-t\sum_n f_n(\theta_0) - N(1-t)\sum_m v_m f_m(\theta_0)}} \\ &= \frac{\int_\Theta h(\theta) e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta}{\int_\Theta e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta} \\ &= \frac{\int_{B_\epsilon^c} h(\theta) e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta}{\int_\Theta e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta} \\ &\quad + \frac{\int_{B_\epsilon} h(\theta) e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta}{\int_\Theta e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta} \\ &:= I_{\epsilon,1}(h) + I_{\epsilon,2}(h) \end{aligned}$$

Writing $\mathbb{E}_{w(v),t} = \mathbb{E}_{w(v),t}[g - \sum_m v_m(f_m - \bar{f})]$, $I_{\epsilon,1} = I_{\epsilon,1}([g - \sum_m v_m(f_m - \bar{f}) - \mathbb{E}_{w(v),t}]^2)$, and similarly for $I_{\epsilon,2}$, we can thus rewrite Eq. (23) as

$$KL(\pi_{w(v)}||\pi) = N^2 \int_0^1 (1-t)\left(I_{\epsilon,1} + I_{\epsilon,2}\right) \mathrm{d}t$$

We first prove that the integral $I_{\epsilon,1}$ over B_ϵ^c is small. Using the above concavity bound, and recalling that $\sum_m v_m = 1$, we have that

$$\begin{aligned} & t\sum_n (f_n(\theta) - f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta) - f_m(\theta_0)) \\ &\quad \leq t\sum_n \left(\nabla_\theta f_n(\theta_0)^T(\theta-\theta_0) - L_0\|\theta-\theta_0\|^2\right) + N(1-t)\sum_m v_m\left(\nabla_\theta f_m(\theta_0)^T(\theta-\theta_0) - L_0\|\theta-\theta_0\|^2\right) \\ &\quad = t\left[\sqrt{N}S_N(\theta_0)^T(\theta-\theta_0) - NL_0\|\theta-\theta_0\|^2\right] + N(1-t)\left[\sum_m v_m \nabla_\theta f_m(\theta_0)^T(\theta-\theta_0) - L_0\|\theta-\theta_0\|^2\right]. \end{aligned}$$

If

$$\sum_m v_m \nabla_\theta f_m(\theta_0) = \frac{S_N(\theta_0)}{\sqrt{N}} \tag{24}$$

then

$$t\sum_n (f_n(\theta) - f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta) - f_m(\theta_0)) \le \sqrt{N} S_N(\theta_0)^T(\theta - \theta_0) - NL_0\|\theta - \theta_0\|^2.$$

Using this bound, we can bound $I_{\epsilon,1}$ by

$$\begin{aligned}
I_{\epsilon,1} &:= \frac{\int_{B_\epsilon^c}[g - \sum_m v_m(f_m - \bar{f}) - \mathbb{E}_{w(v),t}]^2 e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta}{\int_\Theta e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta} \\
&\le \frac{\int_{B_\epsilon^c}[g - \sum_m v_m(f_m - \bar{f}) - \mathbb{E}_{w(v),t}]^2 e^{\sqrt{N}S_N(\theta_0)^T(\theta-\theta_0) - NL_0\|\theta-\theta_0\|^2}\pi_0(\theta)d\theta}{\int_\Theta e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta} \\
&\le \frac{\int_{B_\epsilon^c}[g - \sum_m v_m(f_m - \bar{f})]^2 e^{\sqrt{N}S_N(\theta_0)^T(\theta-\theta_0) - NL_0\|\theta-\theta_0\|^2}\pi_0(\theta)d\theta}{\int_\Theta e^{t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta}.
\end{aligned}$$

Completing the square in the exponent of the numerator, we can rewrite it as

$$\begin{aligned}
&\sqrt{N}S_N(\theta_0)^T(\theta-\theta_0) - NL_0\|\theta-\theta_0\|^2 \\
&= -NL_0\left(\|\theta-\theta_0\|^2 - \frac{S_N(\theta_0)^T(\theta-\theta_0)}{\sqrt{N}L_0}\right) \\
&= -NL_0\left\|(\theta-\theta_0) - \frac{S_N(\theta_0)}{2\sqrt{N}L_0}\right\|^2 + \frac{\|S_N(\theta_0)\|^2}{4L_0}
\end{aligned}$$

We now assume that $\|S_N(\theta_0)\| \le \left(2-\sqrt{2}\right)\epsilon L_0\sqrt{N}$. By Markov's inequality,

$$\begin{aligned}
\Pr\left(\|S_N(\theta_0)\| \ge \left(2-\sqrt{2}\right)\epsilon L_0\sqrt{N}\right) &\le \frac{\mathbb{E}_p\left(\|S_N(\theta_0)\|^2\right)}{\left(2-\sqrt{2}\right)^2\epsilon^2 N L_0^2} \\
&\le 3\frac{\mathbb{E}_p\left(\|\sum_n \nabla_\theta f_n(\theta_0)\|^2\right)}{\epsilon^2 N^2 L_0^2} \\
&= 3\frac{\sum_n \mathbb{E}_p\left(\|\nabla_\theta f_n(\theta_0)\|^2\right)}{\epsilon^2 N^2 L_0^2} \\
&= 3\frac{\mathbb{E}_p\left(\|\nabla_\theta f_1(\theta_0)\|^2\right)}{\epsilon^2 N L_0^2},
\end{aligned}$$

and thus the probability that $\|S_N(\theta_0)\| \le \left(2-\sqrt{2}\right)\epsilon L_0\sqrt{N}$ is bounded from below by $1 - 3\mathbb{E}_p(\|\nabla_\theta f_1(\theta_0)\|^2)/(L_0^2 N\epsilon^2)$. When this event occurs,

$$\begin{aligned}
\left\|(\theta-\theta_0) - \frac{S_N(\theta_0)}{2\sqrt{N}L_0}\right\|^2 &\ge \left(\|\theta-\theta_0\| - \left\|\frac{S_N(\theta_0)}{2\sqrt{N}L_0}\right\|\right)^2 \\
&\ge \left(\|\theta-\theta_0\| - \frac{\left(2-\sqrt{2}\right)\epsilon L_0\sqrt{N}}{2\sqrt{N}L_0}\right)^2 \\
&\ge \left(\|\theta-\theta_0\| - \left(\frac{2-\sqrt{2}}{2}\right)\epsilon\right)^2 \\
&\ge \left(\|\theta-\theta_0\| - \left(\frac{2-\sqrt{2}}{2}\right)\|\theta-\theta_0\|\right)^2 \qquad \text{on } B_\epsilon^c \\
&= \frac{1}{2}\|\theta-\theta_0\|^2,
\end{aligned}$$

and so we can bound $I_{\epsilon,1}$ by

$$I_{\epsilon,1} \leq \frac{e^{\frac{\|S_N(\theta_0)\|^2}{4L_0}} \int_{B_\epsilon^c} [g - \sum_m v_m(f_m - \bar{f})]^2 e^{-NL_0 \left\|(\theta-\theta_0) - \frac{S_N(\theta_0)}{2\sqrt{N}L_0}\right\|^2} \pi_0(\theta) d\theta}{\int_\Theta e^{t\sum_n (f_n(\theta) - f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta) - f_m(\theta_0))] \pi_0(\theta) d\theta}$$
$$\leq \frac{e^{\epsilon^2 NL_0/8} \int_{B_\epsilon^c} [g - \sum_m v_m(f_m - \bar{f})]^2 e^{-NL_0\|\theta-\theta_0\|^2/2} \pi_0(\theta) d\theta}{\int_\Theta e^{t\sum_n (f_n(\theta) - f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta) - f_m(\theta_0))] \pi_0(\theta) d\theta}.$$

To further simplify the upper bound on the numerator, consider the event where

$$\int_{B_\epsilon^c} [g - \sum_m v_m(f_m - \bar{f})]^2 e^{-NL_0\|\theta-\theta_0\|^2/2} \pi_0(\theta) d\theta \leq e^{-NL_0\epsilon^2/4} \int_\Theta \mathbb{E}_p (f_1 - \bar{f})^2(\theta) \pi_0(\theta) d\theta. \tag{25}$$

The probability of Eq. (25) holding can be bounded from below by using Markov's inequality,

$$\Pr\left(\int_{B_\epsilon^c} [g - \sum_m v_m(f_m - \bar{f})]^2 e^{-NL_0\|\theta-\theta_0\|^2/2} \pi_0(\theta) d\theta > e^{-NL_0\epsilon^2/4} \int_\Theta \mathbb{E}_p (f_1 - \bar{f})^2(\theta) \pi_0(\theta) d\theta \right)$$
$$\leq \frac{e^{NL_0\epsilon^2/4} \mathbb{E}_p \left(\int_{B_\epsilon^c} [g - \sum_m v_m(f_m - \bar{f})]^2 e^{-NL_0\|\theta-\theta_0\|^2/2} \pi_0(\theta) d\theta \right)}{\int_\Theta \mathbb{E}_p (f_1 - \bar{f})^2(\theta) \pi_0(\theta) d\theta}. \tag{26}$$

We now use the fact that $(a-b)^2 \leq 2a^2 + 2b^2$ to bound

$$[g - \sum_m v_m(f_m - \bar{f})]^2 = \left[\frac{\sum_n (f_n - \bar{f})}{N} - \sum_m v_m(f_m - \bar{f}) \right]^2$$
$$\leq 2 \left(\frac{\sum_n (f_n - \bar{f})}{N} \right)^2 + 2 \left(\sum_m v_m(f_m - \bar{f}) \right)^2$$
$$\leq 2 \left(\frac{\sum_n (f_n - \bar{f})}{N} \right)^2 + 2 \left(\max_M (f_m - \bar{f}) \sum_m v_m \right)^2$$
$$\leq 2 \left(\frac{\sum_n (f_n - \bar{f})}{N} \right)^2 + 2 \max_M (f_m - \bar{f})^2.$$

Combined with the fact that $\|\theta - \theta_0\| \geq \epsilon$ on B_ϵ^c, we can bound Eq. (26) above by

$$\leq \frac{2 \left(\int_{B_\epsilon^c} \mathbb{E}_p \left[\left(\frac{\sum_n (f_n - \bar{f})}{N} \right)^2 + \max_M (f_m - \bar{f})^2 \right] e^{-NL_0\|\theta-\theta_0\|^2/4} \pi_0(\theta) d\theta \right)}{\int_\Theta \mathbb{E}_p (f_1 - \bar{f})^2(\theta) \pi_0(\theta) d\theta}. \tag{27}$$

Then, by the definition of $\bar{f}$, and the independence of the f_n,

$$\mathbb{E}_p \left[\left(\frac{\sum_n (f_n - \bar{f})}{N} \right)^2 + \max_M (f_m - \bar{f})^2 \right] = \frac{1}{N^2} \sum_n \mathbb{E}_p \left((f_n - \bar{f})^2 \right) + \mathbb{E}_p \left[\max_M (f_m - \bar{f})^2 \right]$$
$$\leq \frac{1}{N} \mathbb{E}_p \left((f_1 - \bar{f})^2 \right) + \mathbb{E}_p \left[\sum_{m=1}^M (f_m - \bar{f})^2 \right]$$
$$= \frac{1}{N} \mathbb{E}_p \left((f_1 - \bar{f})^2 \right) + M \mathbb{E}_p \left((f_1 - \bar{f})^2 \right).$$

Thus, Eq. (27) can be upper bounded by

$$\leq \frac{2 \int_{B_\epsilon^c} e^{-NL_0\|\theta-\theta_0\|^2/4} \mathbb{E}_p (f_1 - \bar{f})^2(\theta) \left(\frac{1}{N} + M \right) \pi_0(\theta) d\theta}{\int_\Theta \mathbb{E}_p (f_1 - \bar{f})^2(\theta) \pi_0(\theta) d\theta}$$
$$\leq \frac{2 \int_{B_\epsilon^c} e^{-NL_0\epsilon^2/4} \mathbb{E}_p (f_1 - \bar{f})^2(\theta) (2M) \pi_0(\theta) d\theta}{\int_\Theta \mathbb{E}_p (f_1 - \bar{f})^2(\theta) \pi_0(\theta) d\theta}$$
$$\leq 4M e^{-NL_0\epsilon^2/4}.$$

This tells us that the probability of Eq. (25) holding can be bounded from below by $1-4Me^{-NL_0\epsilon^2/4}$. We have now obtained an upper bound on the numerator of $I_{\epsilon,1}$ that holds with high probability. To summarize the result so far: if $\sum_m v_m \nabla_\theta f_m(\theta_0) = \frac{S_N(\theta_0)}{\sqrt{N}}$, we have that for any $0 < \epsilon < \epsilon_0$,

$$I_{\epsilon,1} \leq \frac{e^{-\epsilon^2 NL_0/8} \int_\Theta \mathbb{E}_p(f_1 - \bar{f})^2(\theta)\pi_0(\theta)d\theta}{\int_\Theta e^{t\sum_n (f_n(\theta)-f_n(\theta_0))+N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta}.$$

with probability at least

$$1 - 4Me^{-NL_0\epsilon^2/4} - 3\frac{\mathbb{E}_p\left(\|\nabla_\theta f_1(\theta_0)\|^2\right)}{\epsilon^2 NL_0^2}.$$

We continue by finding a lower bound for the denominator of $I_{\epsilon,1}$. We decompose f_n into $f_n^{(1)}$ and $f_n^{(2)}$ and assume that

$$\sum_m v_m(f_m^{(1)}(\theta) - f_m^{(1)}(\theta_0)) - \frac{\sum_n (f_n^{(1)}(\theta) - f_n^{(1)}(\theta_0))}{N} = 0. \tag{28}$$

Then

$$\begin{aligned}
&t\sum_n (f_n(\theta) - f_n(\theta_0)) + N(1-t)\left[\sum_m v_m(f_m(\theta) - f_m(\theta_0))\right] \\
&= \sum_n (f_n(\theta) - f_n(\theta_0)) + N(1-t)\left[\sum_m v_m(f_m(\theta) - f_m(\theta_0)) - \frac{\sum_n (f_n(\theta) - f_n(\theta_0))}{N}\right] \\
&= \sum_n (f_n(\theta) - f_n(\theta_0)) + N(1-t)\left[\sum_m v_m(f_m^{(2)}(\theta) - f_m^{(2)}(\theta_0)) - \frac{\sum_n (f_n^{(2)}(\theta) - f_n^{(2)}(\theta_0))}{N}\right] \\
&= \sum_n (f_n(\theta) - f_n(\theta_0)) + N(1-t)\left[\sum_m v_m f_m^{(2)}(\theta) - \frac{\sum_n f_n^{(2)}(\theta)}{N}\right],
\end{aligned} \tag{29}$$

using Eq. (28) and the fact that $f_n^{(2)}(\theta_0) = 0$ by assumption **A2**. For any $0 < \epsilon' < \epsilon_0$, and $\|\theta - \theta_0\| \leq \epsilon'$, we use assumption **A2** again, along with the fact that $a - b \geq -|a| - |b|$, to bound Eq. (29) from below by

$$\begin{aligned}
&\geq \sum_n (f_n(\theta) - f_n(\theta_0)) - N(1-t)\left|\sum_m v_m f_m^{(2)}(\theta) - \frac{\sum_n f_n^{(2)}(\theta)}{N}\right| \\
&\geq \sum_n (f_n(\theta) - f_n(\theta_0)) - N(1-t)r(\epsilon')\|\theta - \theta_0\|^2\left[\left|\sum_m v_m R(x_m)\right| + \left|\frac{\sum_n R(x_n)}{N}\right|\right] \\
&\geq \sum_n (f_n(\theta) - f_n(\theta_0)) \\
&\quad - N(1-t)r(\epsilon')\|\theta - \theta_0\|^2\left[2\mathbb{E}_p(R(X)) + \left|\sum_m v_m(R(x_m) - \mathbb{E}_p(R(X)))\right| + \frac{|\sum_n (R(x_n) - \mathbb{E}_p(R(X)))|}{N}\right]
\end{aligned} \tag{30}$$

using the fact that $R(x) > 0$ by assumption. Next, we assume that

$$\left|\sum_m v_m(R(x_m) - \mathbb{E}_p(R(X))\right| \leq 1 \qquad \text{and} \qquad \frac{|\sum_n (R(x_n) - \mathbb{E}_p(R(X))|}{N} \leq 1. \tag{31}$$

We can bound the probability of the second inequality holding from below by $1 - \mathrm{Var}_p(R(X))/N$ via Chebychev's inequality:

$$\Pr\left(\left|\frac{1}{N}\sum_n R(x_n) - \mathbb{E}_p(R(X)\right| \geq \frac{\sqrt{\mathrm{Var}_p\left(\frac{1}{N}\sum_n R(x_n)\right)}}{\sqrt{\mathrm{Var}_p\left(\frac{1}{N}\sum_n R(x_n)\right)}}\right) \leq \mathrm{Var}_p\left(\frac{1}{N}\sum_n R(x_n)\right) = \frac{1}{N}\mathrm{Var}_p(R(X)).$$

Substituting the two assumptions from Eq. (31) into Eq. (30) above, we obtain that

$$
\begin{aligned}
&t\sum_n (f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0)) \\
&\qquad \geq \sum_n (f_n(\theta)-f_n(\theta_0)) - 2Nr(\epsilon')\|\theta-\theta_0\|^2\left[\mathbb{E}_p(R(X))+1\right] \\
&\qquad = \sum_n (f_n(\theta)-f_n(\theta_0)) - \gamma Nr(\epsilon')\|\theta-\theta_0\|^2, \quad \text{where} \quad \gamma \coloneqq 2\left[\mathbb{E}_p(R(X))+1\right].
\end{aligned}
$$

Then we can bound the denominator of $I_{\epsilon,1}$ from below by

$$
\begin{aligned}
&\int_\Theta e^{t\sum_n (f_n(\theta)-f_n(\theta_0))+N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))]}\pi_0(\theta)d\theta \\
\geq &\int_{B_{\epsilon'}} e^{t\sum_n (f_n(\theta)-f_n(\theta_0))+N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))]}\pi_0(\theta)d\theta \\
\geq &\int_{B_{\epsilon'}} e^{\sum_n f_n(\theta)-f_n(\theta_0)-\gamma Nr(\epsilon')\|\theta-\theta_0\|^2}\pi_0(\theta)d\theta \\
\geq &\pi_{0,\inf}\int_{B_{\epsilon'}} e^{\sum_n f_n(\theta)-f_n(\theta_0)-\gamma Nr(\epsilon')\|\theta-\theta_0\|^2}d\theta \\
\geq &\pi_{0,\inf}e^{-t_a(\epsilon')}\int_{B_{\epsilon'}} 1_{\sum_n f_n(\theta)-f_n(\theta_0)\geq N(\bar{f}(\theta)-\bar{f}(\theta_0))-t_a} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2}d\theta \qquad (32)
\end{aligned}
$$

$$
\text{where} \quad \pi_{0,\inf} \coloneqq \inf_{\theta\in B_{\epsilon'}}\pi_0(\theta), \quad L_1'(\epsilon') \coloneqq L_1+\gamma r(\epsilon'), \quad \text{and} \quad t_a(\epsilon') \coloneqq \left(\frac{DL_2}{L_1'(\epsilon')a}\right)^{1/2} \text{for } a>0,
$$

where the last line follows from assumption **A3** and the definition of $L_1'(\epsilon')$. The next goal is to bound Eq. (32) from below by

$$
\geq \frac{\pi_{0,\inf}e^{-t_a(\epsilon')}}{2}\int_{B_{\epsilon'}} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2}d\theta \qquad (33)
$$

with high probability. To begin, by Markov's inequality,

$$
\begin{aligned}
&\Pr\left(\int_{B_{\epsilon'}} 1_{\sum_n f_n(\theta)-f_n(\theta_0)\geq N(\bar{f}(\theta)-\bar{f}(\theta_0))-t_a(\epsilon')} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2}d\theta < \frac{\int_{B_{\epsilon'}} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2}d\theta}{2}\right) \\
&= \Pr\left(\int_{B_{\epsilon'}} 1_{\sum_n f_n(\theta)-f_n(\theta_0)< N(\bar{f}(\theta)-\bar{f}(\theta_0))-t_a(\epsilon')} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2}d\theta > \frac{\int_{B_{\epsilon'}} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2}d\theta}{2}\right) \\
&\leq \frac{2\int_{B_{\epsilon'}} \Pr\left(\sum_n f_n(\theta)-f_n(\theta_0) < N(\bar{f}(\theta)-\bar{f}(\theta_0))-t_a(\epsilon')\right)e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2}d\theta}{\int_{B_{\epsilon'}} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2}d\theta}. \qquad (34)
\end{aligned}
$$

We can apply Markov's inequality again to bound the probability inside the integral,

$$
\begin{aligned}
&\Pr\left(\sum_n f_n(\theta)-f_n(\theta_0) < N(\bar{f}(\theta)-\bar{f}(\theta_0))-t_a(\epsilon')\right) \\
=&\Pr\left(\frac{1}{N}\sum_n f_n(\theta_0)-f_n(\theta) - \mathbb{E}_p\left(\frac{1}{N}\sum_n f_n(\theta_0)-f_n(\theta)\right) > \frac{t_a(\epsilon')}{N}\right) \\
\leq&\frac{N^2}{t_a(\epsilon')^2}\mathrm{Var}_p\left(\frac{1}{N}\sum_n f_n(\theta_0)-f_n(\theta)\right) \\
=&\frac{N}{t_a(\epsilon')^2}\mathrm{Var}_p\left(f_1(\theta_0)-f_1(\theta)\right) \\
\leq&\frac{N}{t_a(\epsilon')^2}L_2\|\theta-\theta_0\|^2,
\end{aligned}
$$

using assumption **A3**. Substituting this into Eq. (34) yields

$$\begin{aligned}
&\frac{2\int_{B_{\epsilon'}} \Pr\left(\sum_n f_n(\theta) - f_n(\theta_0) < N(\bar{f}(\theta) - \bar{f}(\theta_0)) - t_a(\epsilon')\right) e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2} d\theta}{\int_{B_{\epsilon'}} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2} d\theta} \\
&\leq \frac{2L_2 \int_{B_{\epsilon'}} N\|\theta-\theta_0\|^2 e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2} d\theta}{t_a(\epsilon')^2 \int_{B_{\epsilon'}} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2} d\theta} \\
&= \frac{L_2}{L_1'(\epsilon')t_a(\epsilon')^2} \frac{\int_{\|u\|\leq\epsilon'\sqrt{2NL_1'(\epsilon')}} \|u\|^2 e^{-\|u\|^2/2} du}{\int_{\|u\|\leq\epsilon'\sqrt{2NL_1'(\epsilon')}} e^{-\|u\|^2/2} du}
\end{aligned} \tag{35}$$

where we make the substitution $u = \sqrt{2NL_1'(\epsilon')}(\theta-\theta_0)$, and cancel the Jacobian terms in numerator and denominator. If $\epsilon' \geq \sqrt{\frac{D}{2NL_1'(\epsilon')}}$, the integral in the denominator can be bounded below by

$$\begin{aligned}
\int_{\|u\|\leq\epsilon'\sqrt{2NL_1'(\epsilon')}} e^{-\|u\|^2/2} du &\geq (2\pi)^{D/2} \int_{\|u\|^2\leq D} (2\pi)^{-D/2} e^{-\|u\|^2/2} du \\
&= (2\pi)^{D/2} \Pr(Y \leq D), \qquad Y \sim \chi_D^2 \\
&\geq \frac{1}{2}(2\pi)^{D/2},
\end{aligned}$$

since the median of a χ_k^2-distribution is $\leq k$. Since $\Theta \subseteq \mathbb{R}^D$, we can upper bound the integral in the numerator by

$$\int_{\|u\|\leq\epsilon'\sqrt{2NL_1'(\epsilon')}} \|u\|^2 e^{-\|u\|^2/2} du \leq (2\pi)^{D/2} \int \|u\|^2 (2\pi)^{-D/2} e^{-\|u\|^2/2} du = D(2\pi)^{D/2},$$

and thus Eq. (35) can be upper bounded by

$$\frac{L_2}{L_1'(\epsilon')t_a(\epsilon')^2} \frac{\int_{\|u\|\leq\epsilon'\sqrt{2NL_1'(\epsilon')}} \|u\|^2 e^{-\|u\|^2/2} du}{\int_{\|u\|\leq\epsilon'\sqrt{2NL_1'(\epsilon')}} e^{-\|u\|^2/2} du} \leq \frac{2L_2 D}{L_1'(\epsilon')t_a(\epsilon')^2} = 2a,$$

using the definition of $t_a(\epsilon')$. Thus with probability at least $1-2a$, we can bound Eq. (32) from below by

$$\begin{aligned}
\text{Eq. (32)} &\geq \frac{\pi_{0,\inf} e^{-t_a(\epsilon')}}{2} \int_{B_{\epsilon'}} e^{-L_1'(\epsilon')N\|\theta-\theta_0\|^2} d\theta \\
&\geq \frac{\pi_{0,\inf} e^{-t_a(\epsilon')}}{2} \left(\frac{1}{2L_1'(\epsilon')N}\right)^{D/2} \int_{\|u\|\leq\epsilon'\sqrt{2L_1'(\epsilon')N}} e^{-\|u\|^2/2} d\theta \\
&\geq \frac{\pi_{0,\inf} e^{-t_a(\epsilon')}}{4} \left(\frac{\pi}{L_1'(\epsilon')N}\right)^{D/2},
\end{aligned}$$

using the same substitution and χ^2 bound as before. This gives us our final lower bound on the denominator of $I_{\epsilon,1}$. To summarize the result at this point: assuming that

$$\sum_m v_m \nabla_\theta f_m(\theta_0) = \frac{S_N(\theta_0)}{\sqrt{N}} \tag{36}$$

$$\sum_m v_m (f_m^{(1)}(\theta) - f_m^{(1)}(\theta_0)) = \frac{\sum_n (f_n^{(1)}(\theta) - f_n^{(1)}(\theta_0))}{N} \tag{37}$$

$$\left|\sum_m v_m (R(x_m) - \mathbb{E}_p(R(X))\right| \leq 1, \tag{38}$$

we have that for any $a > 0$, $\sqrt{\frac{D}{2NL_1'(\epsilon')}} \leq \epsilon' < \epsilon_0$, and $0 \leq \epsilon < \epsilon_0$,

$$I_{\epsilon,1} \leq \frac{e^{-NL_0\epsilon^2/8}\int_\Theta \mathbb{E}_p(f_1-\bar{f})^2(\theta)\pi_0(\theta)d\theta}{\int_\Theta e^{t\sum_n(f_n(\theta)-f_n(\theta_0))+N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0))}\pi_0(\theta)d\theta}$$
$$\leq N^{D/2}e^{t_a(\epsilon')-NL_0\epsilon^2/8}\left(\frac{4}{\pi_{0,\inf}}\int_\Theta \mathbb{E}_p(f_1-\bar{f})^2(\theta)\pi_0(\theta)d\theta\left(\frac{L_1'(\epsilon')}{\pi}\right)^{D/2}\right),$$

with probability at least

$$1 - 4Me^{-NL_0\epsilon^2/4} - 3\frac{\mathbb{E}_p\left(\|\nabla_\theta f_1(\theta_0)\|^2\right)}{\epsilon^2 NL_0^2} - \frac{\mathrm{Var}_p(R(X))}{N} - 2a.$$

We now bound $I_{\epsilon,2}$, corresponding to the integral over B_ϵ. Throughout we assume that the same set of events occur as used in the analysis of $I_{\epsilon,1}$. Note that the denominator of $I_{\epsilon,2}$ is the same as in $I_{\epsilon,1}$, so we apply the same bound; here we focus on the numerator. Since $r(\epsilon)\gamma \leq L_0/2$ by **A2**, we can obtain the following bound for all $\theta \in B_\epsilon$:

$$\begin{aligned}
&t\sum_n(f_n(\theta)-f_n(\theta_0)) + N(1-t)\sum_m v_m(f_m(\theta)-f_m(\theta_0)) \\
&\quad\leq \sum_n(f_n(\theta)-f_n(\theta_0)) + N(1-t)r(\epsilon)\|\theta-\theta_0\|^2\gamma \\
&\quad\leq \sqrt{N}S_N(\theta_0)^T(\theta-\theta_0) - NL_0\|\theta-\theta_0\|^2 + N(1-t)r(\epsilon)\|\theta-\theta_0\|^2\gamma \\
&\quad\leq \sqrt{N}S_N(\theta_0)^T(\theta-\theta_0) - \frac{NL_0}{2}\|\theta-\theta_0\|^2 \\
&\quad= -\frac{NL_0}{2}\left(\|\theta-\theta_0\|^2 - \frac{2S_N(\theta_0)^T(\theta-\theta_0)}{\sqrt{N}L_0}\right) \\
&\quad\leq \frac{\|S_N(\theta_0)\|^2}{2L_0} - \frac{NL_0}{2}\left\|\theta-\theta_0-\frac{S_N(\theta_0)}{\sqrt{N}L_0}\right\|^2.
\end{aligned}$$

Moreover,

$$\begin{aligned}
&g - \sum_m v_m(f_m-\bar{f}) - \mathbb{E}_{w(v),t}[g - \sum_m v_m(f_m-\bar{f})] \\
=&\frac{\sum_n f_n(\theta)-f_n(\theta_0)}{N} - \sum_m v_m(f_m(\theta)-f_m(\theta_0)) - \mathbb{E}_{w(v),t}\left[\frac{\sum_n f_n(\theta)-f_n(\theta_0)}{N} - \sum_m v_m(f_m(\theta)-f_m(\theta_0))\right].
\end{aligned}$$

As before we write, for $\theta \in B_\epsilon$,

$$\frac{\sum_n f_n(\theta)-f_n(\theta_0)}{N} - \sum_m v_m(f_m(\theta)-f_m(\theta_0)) = \frac{\sum_n f_n^{(2)}(\theta)-f_n^{(2)}(\theta_0)}{N} - \sum_m v_m(f_m^{(2)}(\theta)-f_m^{(2)}(\theta_0)),$$

so the right hand side is bounded above by

$$\begin{aligned}
&r(\epsilon)\|\theta-\theta_0\|^2\left[2\mathbb{E}_p(R(X)) + \frac{|\sum_n(R(x_n)-\mathbb{E}_p(R(X))|}{N} + \left|\sum_m v_m((R(x_m)-\mathbb{E}_p(R(X)))\right|\right] \\
&\quad\leq r(\epsilon)\|\theta-\theta_0\|^2\gamma,
\end{aligned}$$

and therefore

$$I_{\epsilon,2} \leq \frac{4r(\epsilon)^2\gamma^2 e^{t_a(\epsilon')}(L_1'(\epsilon')N)^{D/2}\int_{B_\epsilon}\|\theta-\theta_0\|^4 e^{\frac{\|S_N(\theta_0)\|^2}{2L_0} - \frac{NL_0}{2}\left\|\theta-\theta_0-\frac{S_N(\theta_0)}{\sqrt{N}L_0}\right\|^2}\pi_0(\theta)d\theta}{\pi^{D/2}\pi_{0,\inf}}.$$

We further bound the numerator using a similar technique as in the analysis of $I_{\epsilon,1}$:

$$\begin{aligned}\|\theta-\theta_0\|^2 &= \left(\|\theta-\theta_0\| - \frac{\|S_N(\theta_0)\|}{\sqrt{N}L_0} + \frac{\|S_N(\theta_0)\|}{\sqrt{N}L_0}\right)^2 \\ &\le 2\left(\|\theta-\theta_0\| - \frac{\|S_N(\theta_0)\|}{\sqrt{N}L_0}\right)^2 + 2\left(\frac{\|S_N(\theta_0)\|}{\sqrt{N}L_0}\right)^2 \\ &\le 2\left\|(\theta-\theta_0) - \frac{S_N(\theta_0)}{\sqrt{N}L_0}\right\|^2 + 2\left(\frac{\|S_N(\theta_0)\|}{\sqrt{N}L_0}\right)^2.\end{aligned}$$

Note that by Markov's inequality,

$$\forall \zeta > 0, \quad \|S_N(\theta_0)\|^2 \le D/\zeta,$$

with probability $\ge 1 - \zeta \mathbb{E}_p(\|\nabla_\theta f_1(\theta_0)\|^2)/D$. If we further restrict to this event, we have that

$$\begin{aligned}I_{\epsilon,2} &\le \frac{4e^{t_a(\epsilon')}r(\epsilon)^2\gamma^2\pi_{0,\sup}(L_1'(\epsilon')N)^{D/2}}{\pi^{D/2}\pi_{0,\inf}} \int_{B_\epsilon} \|\theta-\theta_0\|^4 e^{\frac{\|S_N(\theta_0)\|^2}{2L_0} - \frac{NL_0}{2}\left\|\theta-\theta_0-\frac{S_N(\theta_0)}{\sqrt{N}L_0}\right\|^2} d\theta \\ &\le \frac{16e^{t_a(\epsilon')}r(\epsilon)^2\gamma^2\pi_{0,\sup}(L_1'(\epsilon')N)^{D/2}e^{\frac{D}{2L_0\zeta}}}{\pi^{D/2}\pi_{0,\inf}} \\ &\qquad \times \int_{B_\epsilon} \left(\left\|(\theta-\theta_0) - \frac{S_N(\theta_0)}{\sqrt{N}L_0}\right\|^2 + \left(\frac{\|S_N(\theta_0)\|}{\sqrt{N}L_0}\right)^2\right)^2 e^{-\frac{NL_0}{2}\left\|\theta-\theta_0-\frac{S_N(\theta_0)}{\sqrt{N}L_0}\right\|^2} d\theta \\ &\le \frac{16e^{t_a(\epsilon')}r(\epsilon)^2\gamma^2\pi_{0,\sup}(L_1'(\epsilon')N)^{D/2}e^{\frac{D}{2L_0\zeta}}(NL_0)^{-(D/2+2)}}{\pi^{D/2}\pi_{0,\inf}} \\ &\qquad \times \int \left(\|u\|^4 + \frac{2\|u\|^2\|S_N(\theta_0)\|^2}{L_0} + \frac{\|S_N(\theta_0)\|^4}{L_0^2}\right) e^{-\frac{1}{2}\|u\|^2} du \\ &\le \frac{16e^{t_a(\epsilon')}r(\epsilon)^2\gamma^2\pi_{0,\sup}(L_1'(\epsilon')N)^{D/2}e^{\frac{D}{2L_0\zeta}}(NL_0)^{-(D/2+2)}(2\pi)^{D/2}}{\pi^{D/2}\pi_{0,\inf}} \\ &\qquad \times \int \left(\|u\|^4 + \frac{2\|u\|^2 D}{L_0\zeta} + \frac{D^2}{L_0^2\zeta^2}\right) (2\pi)^{-D/2} e^{-\frac{1}{2}\|u\|^2} du \\ &\le \frac{16e^{t_a(\epsilon')}D^2r(\epsilon)^2\gamma^2\pi_{0,\sup}(2L_1'(\epsilon')/L_0)^{D/2}e^{\frac{D}{2L_0\zeta}}}{N^2L_0^2\pi_{0,\inf}} \left[3 + \frac{2}{L_0\zeta} + \frac{1}{L_0^2\zeta^2}\right] \\ &\le \frac{16e^{t_a(\epsilon')}D^2r(\epsilon)^2\gamma^2\pi_{0,\sup}(2L_1'(\epsilon')/L_0)^{D/2}e^{\frac{D}{2L_0\zeta}}}{\zeta^2N^2L_0^2\pi_{0,\inf}} \left[3 + \frac{2}{L_0} + \frac{1}{L_0^2}\right], \end{aligned} \tag{39}$$

using the substitution $u = \sqrt{NL_0}\left((\theta-\theta_0) - S_N(\theta_0)/(\sqrt{N}L_0)\right)$. Noting that $t_a(\epsilon') = \Theta(\sqrt{1/a})$, we will set $\sqrt{a} = \zeta$. What remains is to lower bound the probability that the three conditions Eqs. (36) to (38) hold. Denoting by $\psi(x) \in \mathbb{R}^{d_1}$ the coordinates of $f^{(1)}(x,\cdot) - f^{(1)}(x,\theta_0)$ in an orthonormal basis of S_1, we define $Z(x) \in \mathbb{R}^d$, $d \le d_1 + D$, as

$$Z(x) := \begin{bmatrix} \psi(x) \\ \text{indep}\,(\nabla_\theta f(x;\theta_0)) \end{bmatrix}, \tag{40}$$

where indep $(\cdot)$ selects the linearly independent components of the $\nabla_\theta f(x;\theta_0)$ (considered as functions of x). Without loss of generality we can choose $R(x)$ to be linearly independent of $Z(x)$ as a function of x. In this case, the three conditions hold simultaneously if

$$\sum_m v_m\psi_j(x_m) = \frac{\sum_n \psi_j(x_n)}{N} \quad \forall j, \quad \sum_m v_m(R(x_m) - \mathbb{E}_p R(X)) = 0, \quad \sum_m v_m \nabla f_m(\theta_0) = \frac{\sum_n \nabla f_n(\theta_0)}{N}.$$

As in the proof of Theorem 4.1, we can derive that for $\delta \in (0,1]$, if $M \gtrsim J(\delta)^{-1}D(\log N + 1)$, where

$$J(\delta) = \inf_{\substack{a\in\mathbb{R}^{d+1};\\ \|a\|=1}} \Pr\left(\left\langle a, [Z(X), R(X)]^T\right\rangle > \frac{2\mathbb{E}_p(\|Z(X)\|^2)}{\sqrt{N\delta}}\right), \tag{41}$$

then with probability greater than $1-\delta-e^{-J(\delta)M/4}$, Eqs. (36) to (38) hold. We will set $\delta=\zeta^2$. Our particular choice of Z using independent components ensures that $J(\delta)$ is of full rank, and therefore is bounded from below by 0.

We can now obtain a final high probability bound on the KL divergence by combining the bounds on $I_{\epsilon,1}$ and $I_{\epsilon,2}$. In particular, we have that if $M \gtrsim J(\zeta^2)^{-1}D(\log N+1)$, we have that for any $\zeta>0$, $\sqrt{\frac{D}{2NL_1'(\epsilon')}} \le \epsilon' < \epsilon_0$, and $0 \le \epsilon < \epsilon_0$,

$$\begin{aligned}
KL(\pi_{w(v)}||\pi) &= N^2 \int_0^1 (1-t)\left(I_{\epsilon,1}+I_{\epsilon,2}\right) \mathrm{d}t \\
&\le N^{D/2+2} e^{\frac{1}{\zeta}\sqrt{\frac{DL_2}{L_1'(\epsilon')}} - NL_0\epsilon^2/8} \left(\frac{2}{\pi_{0,\inf}} \int_\Theta \mathbb{E}_p(f_1-\bar{f})^2(\theta)\pi_0(\theta)d\theta \left(\frac{L_1'(\epsilon')}{\pi}\right)^{D/2}\right) \\
&\quad + \frac{r(\epsilon)^2 e^{\frac{1}{\zeta}\left(\sqrt{\frac{DL_2}{L_1'(\epsilon')}}+\frac{D}{2L_0}\right)}}{\zeta^2} \left(\frac{8D^2\gamma^2\pi_{0,\sup}(2L_1'(\epsilon')/L_0)^{D/2}}{L_0^2\pi_{0,\inf}} \left[3+\frac{2}{L_0}+\frac{1}{L_0^2}\right]\right)
\end{aligned}$$

with probability at least

$$1-e^{-J(\zeta^2)M/4} - 4Me^{-NL_0\epsilon^2/4} - 3\frac{\mathbb{E}_p\left(\|\nabla_\theta f_1(\theta_0)\|^2\right)}{\epsilon^2 NL_0^2} - \frac{\mathrm{Var}_p(R(X))}{N} - 3\zeta^2 - \zeta\frac{\mathbb{E}_p(\|\nabla_\theta f_1(\theta_0)\|^2)}{D}.$$

If $\zeta^{-1}=o(-\log r(\epsilon))$ as $\epsilon\to 0$, then the second term in the KL bound converges to 0. Consequently if $\log N + |\log r(\epsilon)| = o(N\epsilon^2)$ as $N\to\infty$, the KL divergence converges to 0 as $N\to\infty$. Similarly, $J(\zeta^2)$ converges to a nonzero constant as long as $\zeta\sqrt{N}\to\infty$. Both conditions are possible to satisfy, for any $r(\epsilon)$, by making $\epsilon\to 0$ slowly enough as a function of N.

□

B.3 Proof of Theorem 4.3

Proof. For step size $\gamma\in[0,1]$, the result $\hat{w}_{k+1}$ of the approximate Newton step prior to projection onto $\mathcal{W}$ is

$$\begin{aligned}
\hat{w}_{k+1} &= w_k + \gamma(G(w_k)+\tau I)^{-1}H(w_k)(1-w_k) \\
&= w_k + \gamma(G(w_k)+\tau I)^{-1}H(w_k)(w_k^\star - w_k) + \gamma(G(w_k)+\lambda I)^{-1}H(w_k)(1-w_k^\star) \\
&= w_k + \gamma(G(w_k)+\tau I)^{-1}G(w_k)(w_k^\star - w_k) + \gamma(G(w_k)+\lambda I)^{-1}H(w_k)(1-w_k^\star),
\end{aligned}$$

where the last line follows from the fact that $w_k, w_k^\star \in \mathcal{W}$. Consider the distance $\left\|w_{k+1}-w_{k+1}^\star\right\|$. Since $w_{k+1}^\star$ is the projection of w_{k+1} onto a convex set $\mathcal{W}^\star$, we have that

$$\left\|w_{k+1}-w_{k+1}^\star\right\| \le \left\|w_{k+1}-w_k^\star\right\|.$$

Furthermore, since w_{k+1} is the projection of $\hat{w}_{k+1}$ onto $\mathcal{W}$, and $w_k^\star\in\mathcal{W}$,

$$\left\|w_{k+1}-w_k^\star\right\| \le \left\|\hat{w}_{k+1}-w_k^\star\right\|.$$

Then, we have that:

$$\begin{aligned}
\|\hat{w}_{k+1}-w_k^\star\| &= \left\|\left(I-\gamma(G(w_k)+\tau I)^{-1}G(w_k)\right)(w_k-w_k^\star) + \gamma(G(w_k)+\lambda I)^{-1}H(w_k)(1-w_k^\star)\right\| \\
&\le \left\|\left(I-\gamma(G(w_k)+\tau I)^{-1}G(w_k)\right)(w_k-w_k^\star)\right\| + \gamma(\epsilon\|w_k-w_k^\star\|+\delta) \\
&\le \eta\left\|w_k-w_k^\star\right\| + \gamma\delta.
\end{aligned}$$

The first bound follows from our assumption in the theorem statement, and the triangle inequality. The second bound follows via the following logic. Since $\pi_w \propto \exp(w^T f_{[M]}(\theta))\pi_0(\theta)$, it has the same support for all $w\in\mathcal{W}$. Since $G(w)$ is the covariance matrix of $f_{[M]}(\theta)$, the null space of $G(w)$ is spanned by the set of vectors $u\in\mathbb{R}^M$ such that $u^T f_{[M]}(\theta)=c$ for some constant $c\in\mathbb{R}$ for all θ within the support of π_w. Therefore the null space of $G(w)$ is the same for all $w\in\mathcal{W}$. Since $w^T f_{[M]} = (w+u)^T f_{[M]}$ (up to a constant in θ) for any $w\in\mathcal{W}$ and u in the null space of $G(w)$, we can augment any closed convex subset $\mathcal{W}^\star \subseteq \arg\min_{w\in\mathcal{W}} \mathrm{KL}(\pi_w||\pi)$ with

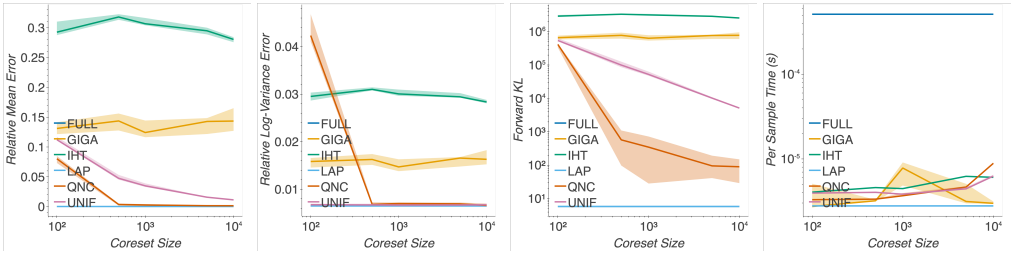


Figure 2: Relative mean and log-variance error, forward KL divergence and per sample time to sample from the respective posteriors for the synthetic Gaussian experiment. Our algorithm (QNC) consistently provides an improvement in coreset quality over the other subsampling methods. All methods provide a significant reduction in time to sample from the respective posteriors.

the null space of $G(w)$ to form $\mathcal{W}_2^\star = \{w + u : w \in \mathcal{W}^\star,\, u \in \operatorname{null} G(w), w + u \in \mathcal{W}\}$, which is still a closed convex subset of $\mathcal{W}^\star \subseteq \arg\min_{w\in\mathcal{W}} \mathrm{KL}(\pi_w||\pi)$. Therefore any maximal convex subset $\mathcal{W}^\star$ must be of the previous form augmented with the null space of $G(w)$. Because $w_k^\star$ is a projection, $w_k^\star - w_k$ must be orthogonal to this null space. Denoting the full eigendecomposition $G(w) = V\Lambda(w)V^T$, $V \in \mathbb{R}^{M\times M}$, $\Lambda(w) \in \mathbb{R}^{M\times M}$, including zero eigenvalues, and similarly the reduced eigendecomposition $G(w) = V_+\Lambda_+(w)V_+^T$ with the null space removed,

$$\begin{aligned}
\left\| \left(I - \gamma(G(w_k) + \tau I)^{-1} G(w_k)\right)(w_k - w_k^\star) \right\| &= \left\| V\left(I - \gamma(\Lambda(w_k) + \tau)^{-1}\Lambda(w_k)\right) V^T(w_k - w_k^\star) \right\| \\
&= \left\| V_+\left(I - \gamma(\Lambda_+(w_k) + \tau)^{-1}\Lambda_+(w_k)\right) V_+^T(w_k - w_k^\star) \right\| \\
&\leq (1 - \gamma\xi) \left\| V_+^T(w_k - w_k^\star) \right\| \\
&= (1 - \gamma\xi) \left\| V^T(w_k - w_k^\star) \right\| \\
&= (1 - \gamma\xi) \left\| (w_k - w_k^\star) \right\| .
\end{aligned}$$

Finally solving the earlier recursion,

$$\begin{aligned}
\|w_k - w_k^\star\| &\leq \eta^k \|w_0 - w_0^\star\| + \gamma\delta \sum_{j=0}^{k-1} \eta^j \\
&= \eta^k \|w_0 - w_0^\star\| + \gamma\delta \left(\frac{1 - \eta^k}{1 - \eta}\right).
\end{aligned}$$

□

C Experiments

In this section, we present some additional results, along with the full details of the Bayesian radial basis function regression experiment. We also discuss the use of a Laplace approximation for the low-cost approximation $\hat{\pi}$ needed for the sparse regression methods: greedy iterative geodesic ascent (GIGA) and iterative hard thresholding (IHT). Finally, we include a note on the overall performance of those methods.

C.1 Synthetic Gaussian location model

From Fig. 2 we see that our additional results match closely those in Section 5. Our method (QNC) consistently outperforms the other subsampling methods, while the Laplace approximation gives the best results, by design.

C.2 Bayesian sparse linear regression

From Fig. 3 we see that our additional results match closely those in Section 5. Our method (QNC) consistently outperforms the other methods. In this experiment, the prior is heavy tailed, whilst the likelihood has sub-exponential tails. However, when the data is in fact concentrated on a particular subspace of the overall space, then the posterior can have heavy tails, even when the number of data

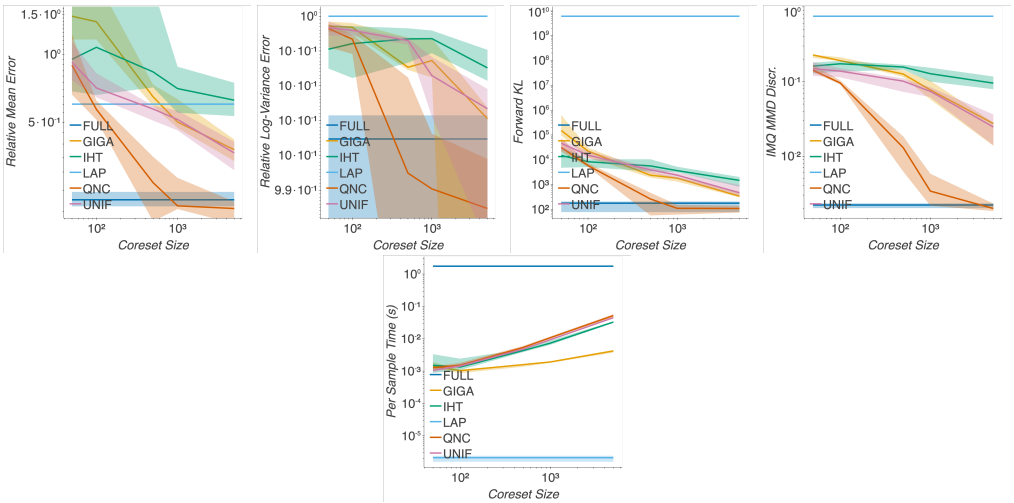


Figure 3: Relative mean and log-variance error, forward KL divergence, IMQ maximum mean discrepancy (MMD) and per sample time to sample from the respective posteriors for the sparse regression experiment. Our algorithm (QNC) consistently provides an improvement in coreset quality over the other methods. All methods provide a significant reduction in time to sample from the respective posteriors. The poor performance of the Laplace approximation in this heavy-tailed example can be seen particularly clearly in the forward KL and IMQ MMD plots.

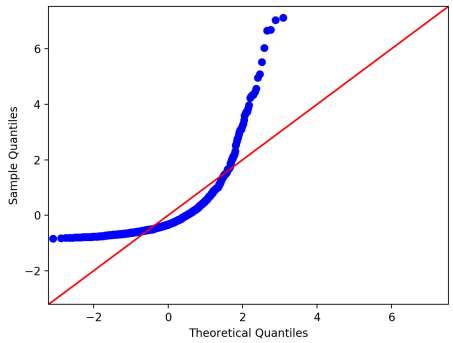


Figure 4: Quantile-quantile plot, comparing the quantiles of the (centred and scaled) marginal posterior distribution of λ_{13} to the quantiles of a standard normal distribution.

points is large. We can clearly see this in Fig. 4, where we plot the quantiles of the (centred and scaled) marginal posterior distribution of the parameter λ_{13} to the quantiles of a standard normal distribution. This parameter is strictly positive, but we see that the right hand tail of the distribution is significantly heavier than a normal distribution.

In the forward KL and IMQ maximum mean discrepancy (MMD) plots, we can see clearly that the Laplace approximation is performing poorly, as it is underestimating the tails of the posterior distribution. Calculating the IMQ stein discrepancy in this experiment is computationally intractable with the size of dataset we consider.

C.3 Heavy-tailed Bayesian logistic regression

From Fig. 5 we see that our additional results match closely those in Section 5. Our method (QNC) consistently outperforms the other methods.

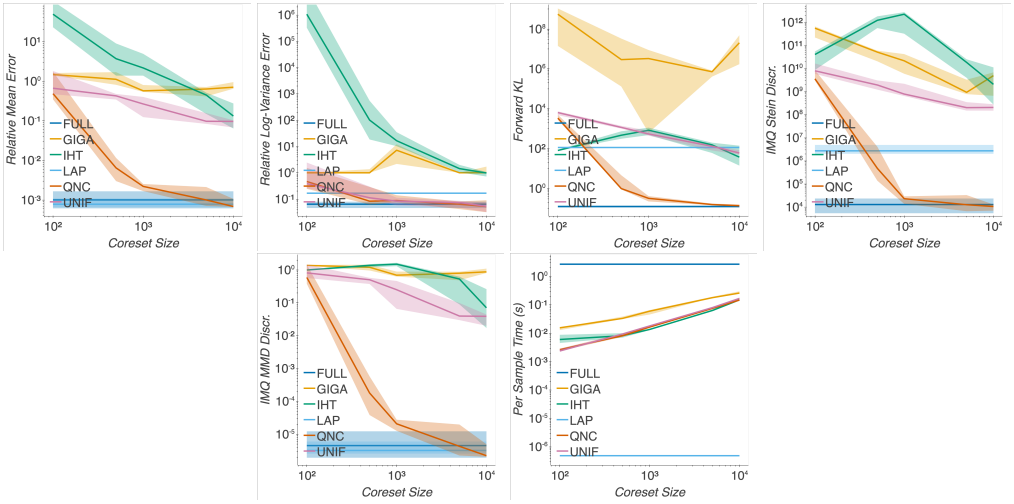


Figure 5: Relative mean and log-variance error, forward KL divergence, IMQ stein discrepancy, IMQ maximum mean discrepancy (MMD) and per sample time to sample from the respective posteriors for the sparse regression experiment. Our algorithm (QNC) generally provides an improvement in coreset quality over the other methods. All methods provide a significant reduction in time to sample from the respective posteriors.

C.4 Bayesian radial basis function regression

Our final comparison is on a Bayesian basis function regression example, and we provide the full details of that experiment here. We perform inference for the coefficients $\alpha \in \mathbb{R}^D$ in a linear combination of radial basis functions $b_k(x) = \exp\left(-1/2\sigma_k^2 \left(x - \mu_k\right)^2\right), k = 1, \ldots, D,$

$$y_n = b_n^T \alpha + \epsilon_n, \quad \epsilon_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right), \quad b_n = \left[\begin{array}{ccc} b_1\left(x_n\right) & \cdots & b_D\left(x_n\right) \end{array}\right]^T, \quad \alpha \sim \mathcal{N}\left(\mu_0, \sigma_0^2 I\right).$$

The data consists of $N = 100,000$ records of house sale log-price $y_n \in \mathbb{R}$ as a function of latitude / longitude coordinates $x_n \in \mathbb{R}^2$ in the UK. [3] In order to perform inference we generate 50 basis functions for each of 6 scales $\sigma_k \in \{0.2, 0.4, 0.8, 1.2, 1.6, 2.0\}$ by generating means μ_k uniformly from the data, and including a basis with scale 100 (i.e. nearly constant) and mean corresponding to the mean latitude and longitude of the data. Thus, we have 301 total basis functions, and so $D = 301$. The prior and noise parameters $\mu_0, \sigma_0^2, \sigma^2$ are equal to the empirical mean, second moment, and variance of the price paid $(y_n)_{n=1}^N$ across the whole dataset, respectively. Closed form expressions are available for the subsampled posterior distributions [21, Appendix B], and we can sample from them without Monte Carlo.

From Fig. 6 we see that our additional results match closely those in Section 5. Our method (QNC) generally outperforms the other methods, but has more trouble capturing the variance of the posterior, compared to the other subsampling approaches. Though the Laplace approximation captures the mean of the posterior well, it does not approximated the variance well, leading to its overall poor performance.

C.5 Sparse regression methods with Laplace approximation

In Section 5, we use a uniformly sampled coreset approximation of size M as the low-cost approximation $\hat{\pi}$ for GIGA and IHT. Here, we include additional results with a Laplace approximation used for $\hat{\pi}$. These are labeled as GIGA-LAP and IHT-LAP respectively. In Fig. 7, we see that in the

[3] This dataset was constructed by merging housing prices from the UK land registry data `https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads` with latitude & longitude coordinates from the Geonames postal code data `http://download.geonames.org/export/zip/`. The housing price dataset contains HM Land Registry data © Crown copyright and database right 2021. This data is licensed under the Open Government Licence v3.0. The postal code data is licensed under a Creative Commons Attribution 4.0 License.

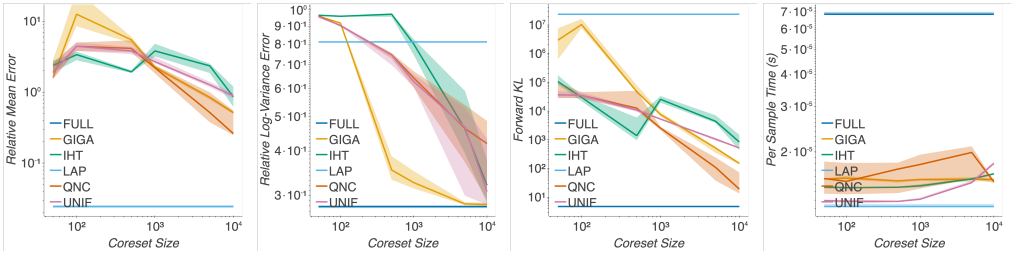
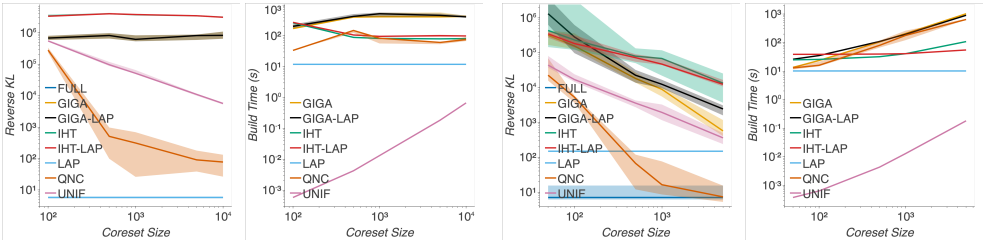
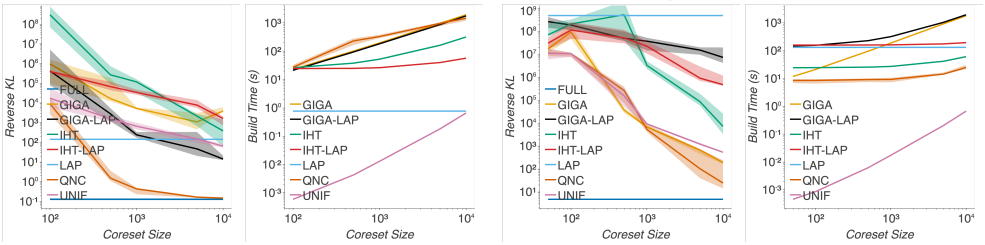


Figure 6: Relative mean and log-variance error, forward KL divergence and per sample time to sample from the respective posteriors for the basis regression experiment. Our algorithm (QNC) generally provides an improvement in coreset quality over the other methods. All methods provide a significant reduction in time to sample from the respective posteriors. The poor performance of the Laplace approximation in this high-dimensional example can be seen particularly clearly in the forward KL and log-variance plots.



(a) Gaussian

(b) Sparse linear regression



(c) Logistic regression

(d) Basis function regression

Figure 7: Reverse KL divergence (left) and build time in seconds (right) for each experiment, with added plots for GIGA and IHT with a Laplace approximation used for $\hat{\pi}$. We plot the median and a shaded area between the 25$^{\text{th}}$/75$^{\text{th}}$ percentiles over 10 random trials. Using a Laplace approximation used for $\hat{\pi}$ rarely provides an improvement in performance, and in some cases performs significantly worse than with the uniform choice of $\hat{\pi}$.

synthetic Gaussian experiment, the choice of $\hat{\pi}$ has very little effect, and both GIGA and IHT perform quite badly in this large, high-dimensional example. In the sparse regression and basis function regression experiments, using a Laplace approximation for $\hat{\pi}$ leads to a worse performance. This may be unsurprising because the Laplace approximation (LAP) does not provide a good approximation to the true posterior in these settings, as detailed above and in Section 5. Only in the logistic regression experiment do we see an improvement in performance from GIGA-LAP over GIGA.

One reason for this is that the Laplace approximation provides limited help in coreset construction. Due to the large data set sizes used in our experiments, we only use a small number of samples ($S = 500$) in the coreset construction process. However, especially in high dimensions, we need a large number of samples to obtain a good L_2 approximation to the posterior from the Laplace approximation.

C.6 Performance of sparse regression methods

Throughout our experiments, we see that the sparse regression methods GIGA and IHT perform poorly. One reason for this is that these methods involve approximating a high dimensional integral with a small, fixed, number of samples. This problem persists no matter which choice of low-cost

approximation $\hat{\pi}$ we make, and is particularly problematic in high dimensions [33]. As noted by [21], methods like these are further limited by using a fixed approximation $\hat{\pi}$, rather than iteratively updating it as in the sparse variational inference approach.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Fast Bayesian Coresets via Subsampling and Quasi-Newton Refinement
Publication Status	□Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Submitted to the International Conference on Machine Learning, 2022.

Student Confirmation

Student Name:	Cian Naik		
Contribution to the Paper	I am the first author of this paper. I have contributed to designing and implementing the algorithm, running the experiments, developing the theoretical results and writing the paper.		
Signature	Cian Naik	Date	25/03/22

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor François Caron		
Supervisor comments I certify that Cian made a substantial contribution to the publication, as described above.		
Signature	Date	28/03/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

5 Conclusion

Contents

5.1 Summary **145**
5.2 Future Directions **146**

In this chapter, we conclude by summarising the work contained in the preceding chapters of this thesis, as well as outlining potential directions for future work based upon them.

5.1 Summary

In this thesis, we present methodological and theoretical contributions to Bayesian statistics. In particular, we focus on using Bayesian methods to uncover latent structure within a dataset.

Chapters 2 and 3 are concerned with Bayesian nonparametric models for statistical network analysis. Based on the framework of Caron and Fox (2017), we develop two novel models. The first of these is a model for sparse, static networks with core-periphery structure. We introduce a class of sparse networks with this structure, provide theoretical results on their sparsity properties, and provide methods to simulate from them. We also introduce a posterior inference

algorithm, and use it to show that our method compares favourably against contemporary alternatives.

The second model we develop is for sparse dynamic networks. Here, we model the evolution of sparse networks over time, where edges can appear and disappear, and the popularities of the nodes may also change. We also introduce an approximate posterior inference algorithm that allows us to capture the long-term evolution of the node popularities in various real datasets.

In Chapter 4, we study the problem of data subsampling as a preprocessing step to Bayesian inference. The idea we utilize is that, in a large scale data setting, most of the data may be redundant, with a small, weighted subset - a (Bayesian) *coreset* - sufficing to capture the whole dataset in some sense. Our contribution is a novel method for Bayesian coreset construction. The key insight in our work is that we can form a coreset by first uniformly sampling a subset of data, and then optimizing the weights on those points using a novel quasi-Newton method. We provide theoretical results to show that our method finds a high-quality coreset with high probability, and empirical results to show that it compares favourably against realistic alternatives.

5.2 Future Directions

When developing the methodological approaches contained in this thesis, we also discover some directions for future extensions to our work. We briefly discuss some of these here.

In Chapter 2, a necessary property of our model is that the relative size of the dense core is vanishingly small, in order for the overall network to be sparse. A potentially useful extension to our work would be to consider the case where the overall network is dense, with a sparse periphery region but a core with a relative size that is bounded above zero. Furthermore, whilst our model allows for the existence of multiple overlapping communities, we are concerned with modelling networks with a single core. An alternative approach proposed by Kojaku and Masuda (2017)

is to allow for the possibility multiple core-periphery pairs within the same network. Extending our framework to allow for this could add some useful flexibility.

A related model to the one we develop in Chapter 3 is that of Miscouridou et al. (2018), who use mutually-exciting Hawkes processes to model temporal interaction data. Their work incorporates additional structure by building on that of Todeschini et al. (2020), in order to incorporate community structure to the network. Exploring how communities appear, evolve and merge could have many practical uses. Thus, extending our model in order to capture both evolving node popularity and dynamically changing community structure could expand the applications of our work. Furthermore, our model assumes that the time between observations of the network is constant. If in fact these time intervals are not constant, it may be helpful to use a continuous-time version of our model. This could be done by considering a birth-death process for the interactions between nodes, where each interaction has a certain lifetime distribution. The continuously evolving node sociabilities could then by described by the Dawson-Watanabe superprocess (Watanabe, 1968; Dawson, 1975).

As with all current Bayesian coreset approaches, the construction method we develop in Chapter 4 has so far only been evaluated on relatively simple models, such as logistic regression and sparse linear regression. An immediate extension to this would be to investigate the performance of our method in more complex cases such as mixture or network models. Furthermore, when considering the computational trade-offs between our method and simply running MCMC on the full dataset, the fact remains that both approaches have a linear time dependence on the size of the (full) dataset. In order to make our work more applicable to the very large data regime, it would be valuable to investigate whether sublinear time compression can be achieved, whilst retaining some theoretical performance guarantees.

Bibliography

Aalen, O. (1992), 'Modelling heterogeneity in survival analysis by the compound Poisson distribution', *The Annals of Applied Probability* pp. 951–972.

Agarwal, P., Har-Peled, S. and Varadarajan, K. (2005), 'Geometric approximation via coresets', *Combinatorial and computational geometry* **52**.

Ahn, S., Korattikara, A. and Welling, M. (2012), Bayesian posterior sampling via stochastic gradient Fisher scoring, *in* 'International Conference on Machine Learning'.

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

Airoldi, E. M., Costa, T. B. and Chan, S. H. (2014), Stochastic blockmodel approximation of a graphon: Theory and consistent estimation, *in* 'NIPS', Vol. 26.

Albert, R. and Barabási, A.-L. (2002), 'Statistical mechanics of complex networks', *Reviews of modern physics* **74**(1), 47.

Aldous, D. J. (1981), 'Representations for partially exchangeable arrays of random variables', *Journal of Multivariate Analysis* **11**(4), 581–598.

Arthur, D. and Vassilvitskii, S. (2006), k-means++: The advantages of careful seeding, Technical report, Stanford.

Bachem, O., Lucic, M., Hassani, S. H. and Krause, A. (2016), Approximate k-means++ in sublinear time, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 30.

Bachem, O., Lucic, M. and Krause, A. (2015), Coresets for nonparametric estimation-the case of dp-means., *in* 'ICML', pp. 209–217.

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

Bardenet, R., Doucet, A. and Holmes, C. (2017), 'On markov chain monte carlo methods for tall data', *The Journal of Machine Learning Research* **18**(1), 1515–1557.

Bertoin, J., Fujita, T., Roynette, B. and Yor, M. (2006), 'On a particular class of self-decomposable random variables : the durations of Bessel excursions straddling independent exponential times', *Probability and Mathematical Statistics* **26**(2), 315–366.

Bickel, P. J. and Chen, A. (2009), 'A nonparametric view of network models and Newman–Girvan and other modularities', *Proceedings of the National Academy of Sciences* **106**(50), 21068–21073.

Bickel, P. J., Chen, A. and Levina, E. (2011), 'The method of moments and degree distributions for network models', *The Annals of Statistics* **39**(5), 2280–2301.

Bishop, C. M. (2006), *Pattern recognition and machine learning*, springer.

Bollobás, B. and Riordan, O. (2009), Metrics for sparse graphs, *in* S. Huczynska, J. Mitchell and C. Roney-Dougal, eds, 'Surveys in combinatorics', Vol. 365 of *London Mathematical Society Lecture Note Series*, Cambridge University Press, arXiv:0708.1919, pp. 211–287.

Brix, A. (1999), 'Generalized gamma measures and shot-noise Cox processes', *Advances in Applied Probability* **31**(4), 929–953.

Broido, A. D. and Clauset, A. (2019), 'Scale-free networks are rare', *Nature communications* **10**(1), 1–10.

Campbell, T. and Beronov, B. (2019), Sparse variational inference: Bayesian coresets from scratch, *in* 'Advances in Neural Information Processing Systems', pp. 11461–11472.

Campbell, T. and Broderick, T. (2018), 'Bayesian coreset construction via greedy iterative geodesic ascent', *arXiv preprint arXiv:1802.01737* .

Campbell, T. and Broderick, T. (2019), 'Automated scalable bayesian inference via hilbert coresets', *The Journal of Machine Learning Research* **20**(1), 551–588.

Caron, F. and Fox, E. (2017), 'Sparse graphs using exchangeable random measures', *Journal of the Royal Statistical Society B* **79**, 1–44. Part 5.

Caron, F., Panero, F. and Rousseau, J. (2017), 'On sparsity, power-law and clustering properties of graphex processes', *arXiv* pp. arXiv–1708.

Caron, F., Teh, Y. and Murphy, T. (2014), 'Bayesian nonparametric plackett-luce models for the analysis of preferences for college degree programmes', *The Annals of Applied Statistics* **8**(2), 1145–1181.

Clauset, A., Shalizi, C. R. and Newman, M. E. J. (2009), 'Power-law distributions in empirical data', *SIAM review* **51**(4), 661–703.

Csermely, P., London, A., Wu, L.-Y. and Uzzi, B. (2013), 'Structure and dynamics of core/periphery networks', *Journal of Complex Networks* **1**(2), 93–123.

Daley, D. J. and Vere-Jones, D. (2008), *An Introduction to the Theory of Point Processes. Volume II: General Theory and Structure*, Springer.

Dawson, D. A. (1975), 'Stochastic evolution equations and related measure processes', *Journal of Multivariate Analysis* **5**(1), 1–52.

Devroye, L. and James, L. F. (2014), 'On simulation and properties of the stable law', *Statistical Methods and Applications* **23**(3), 307–343.

Diaconis, P. and Janson, S. (2008), 'Graph limits and exchangeable random graphs', *Rendiconti di Matematica e delle sue Applicazioni. Serie VII* pp. 33–61.

Efron, B. (1992), Bootstrap methods: another look at the jackknife, *in* 'Breakthroughs in statistics', Springer, pp. 569–593.

Efron, B. and Tibshirani, R. J. (1994), *An introduction to the bootstrap*, CRC press.

Epifani, I. and Lijoi, A. (2010), 'Nonparametric priors for vectors of survival functions', *Statistica Sinica* pp. 1455–1484.

Erdös, P. and Rényi, A. (1959), 'On random graphs', *Publicationes Mathematicae* **6**, 290–297.

Feldman, D., Faulkner, M. and Krause, A. (2011), Scalable training of mixture models via coresets, *in* 'Advances in neural information processing systems', pp. 2142–2150.

Feldman, D. and Langberg, M. (2011), A unified framework for approximating and clustering data, *in* 'Proceedings of the forty-third annual ACM symposium on Theory of computing', pp. 569–578.

Feldman, D., Schmidt, M. and Sohler, C. (2020), 'Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering', *SIAM Journal on Computing* **49**(3), 601–657.

Frank, M., Wolfe, P. et al. (1956), 'An algorithm for quadratic programming', *Naval research logistics quarterly* **3**(1-2), 95–110.

Gao, C., Lu, Y. and Zhou, H. H. (2015), 'Rate-optimal graphon estimation', *The Annals of Statistics* **43**(6), 2624–2652.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D., Vehtari, A. and Rubin, D. (2014), *Bayesian Data Analysis*, Chapman and Hall/CRC.

Ghosal, S. and Van der Vaart, A. (2017), *Fundamentals of nonparametric Bayesian inference*, Vol. 44, Cambridge University Press.

Griffin, J. E., Kolossiatis, M. and Steel, M. F. (2013), 'Comparing distributions by using dependent normalized random-measure mixtures', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **75**(3), 499–529.

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**(2), 525–545.

Hjort, N. L., Holmes, C., Müller, P. and Walker, S. G. (2010), *Bayesian nonparametrics*, Vol. 28, Cambridge University Press.

Hoff, P. D. (2009), 'Multiplicative latent factor models for description and prediction of social networks', *Computational and Mathematical Organization Theory* **15**(4), 261–272.

Hoff, P. D., Raftery, A. E. and Handcock, M. S. (2002), 'Latent space approaches to social network analysis', *Journal of the american Statistical association* **97**(460), 1090–1098.

Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. (2013), 'Stochastic variational inference', *Journal of Machine Learning Research* **14**, 1303–1347.
URL: *http://jmlr.org/papers/v14/hoffman13a.html*

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

Holme, P. (2005), 'Core-periphery organization of complex networks', *Physical Review E* **72**(4), 046111.

Holme, P. (2019), 'Rare and everywhere: Perspectives on scale-free networks', *Nature communications* **10**(1), 1–3.

Hoover, D. N. (1979), 'Relations on probability spaces and arrays of random variables', *Preprint, Institute for Advanced Study, Princeton, NJ* .

Hougaard, P. (1986), 'Survival models for heterogeneous populations derived from stable distributions', *Biometrika* **73**(2), 387–396.

Huggins, J. H., Campbell, T. and Broderick, T. (2016), 'Coresets for scalable bayesian logistic regression', *arXiv preprint arXiv:1605.06423* .

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

Jia, J. and Benson, A. R. (2019), Random spatial network models for core-periphery structure, *in* 'Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining', pp. 366–374.

Kallenberg, O. (2005), *Probabilistic symmetries and invariance principles*, Springer.

Kim, M. and Leskovec, J. (2012), 'Multiplicative attribute graph model of real-world networks', *Internet Mathematics* **8**(1-2), 113–160.

Kingman, J. (1967), 'Completely random measures', *Pacific Journal of Mathematics* **21**(1), 59–78.

Kojaku, S. and Masuda, N. (2017), 'Finding multiple core-periphery pairs in networks', *Physical Review E* **96**(5), 052313.

Korattikara, A., Chen, Y. and Welling, M. (2014), Austerity in MCMC land: cutting the Metropolis-Hastings budget, *in* 'International Conference on Machine Learning'.

Lauritzen, S. (2008), 'Exchangeable Rasch matrices', *Rendiconti di Matematica, Serie VII* **28**, 83–95.

Lee, J., James, L. F. and Choi, S. (2016), Finite-dimensional bfry priors and variational bayesian inference for power law models, *in* 'Advances in Neural Information Processing Systems', pp. 3162–3170.

Lee, M.-L. T. and Whitmore, G. A. (1993), 'Stochastic processes directed by randomized time', *Journal of applied probability* pp. 302–314.

Leisen, F. and Lijoi, A. (2011), 'Vectors of two-parameter poisson–dirichlet processes', *Journal of Multivariate Analysis* **102**(3), 482–495.

Leisen, F., Lijoi, A. and Spanó, D. (2013), 'A vector of dirichlet processes', *Electronic Journal of Statistics* **7**, 62–90.

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**(4), 715–740.

Lijoi, A., Nipoti, B. and Prünster, I. (2014), 'Bayesian inference with dependent normalized completely random measures', *Bernoulli* **20**(3), 1260–1291.

Lijoi, A. and Prünster, I. (2010), Models beyond the Dirichlet process, *in* P. M. S. G. W. N. L. Hjort, C. Holmes, ed., 'Bayesian Nonparametrics', Cambridge University Press.

Lloyd, J., Orbanz, P., Ghahramani, Z. and Roy, D. (2012), Random function priors for exchangeable arrays with applications to graphs and relational data, *in* 'NIPS', Vol. 25, pp. 1007–1015.

Lucic, M., Bachem, O. and Krause, A. (2016), Strong coresets for hard and soft bregman clustering with applications to exponential family mixtures, *in* 'Artificial intelligence and statistics', PMLR, pp. 1–9.

Lucic, M., Faulkner, M., Krause, A. and Feldman, D. (2017*a*), 'Training gaussian mixture models at scale via coresets', *The Journal of Machine Learning Research* **18**(1), 5885–5909.

Lucic, M., Faulkner, M., Krause, A. and Feldman, D. (2017*b*), 'Training mixture models at scale via coresets', *stat* **1050**, 23.

Maclaurin, D. and Adams, R. (2014), Firefly Monte Carlo: exact MCMC with subsets of data, *in* 'Conference on Uncertainty in Artificial Intelligence'.

Manousakas, D., Xu, Z., Mascolo, C. and Campbell, T. (2020), Bayesian pseudocoresets, *in* 'Advances in Neural Information Processing Systems'.

Miller, K., Griffiths, T. and Jordan, M. (2009), Nonparametric latent feature models for link prediction, *in* 'NIPS'.

Miscouridou, X., Caron, F. and Teh, Y. W. (2018), Modelling sparsity, heterogeneity, reciprocity and community structure in temporal interaction data, *in* 'Advances in Neural Information Processing Systems', pp. 2343–2352.

Newman, M. (2009), *Networks: an introduction*, OUP Oxford.

Newman, M. (2018), *Networks*, Oxford university press.

Nowicki, K. and Snijders, T. (2001), 'Estimation and prediction for stochastic blockstructures', *Journal of the American Statistical Association* **96**(455), 1077–1087.

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**(2), 437–461.

Politis, D. N., Romano, J. P. and Wolf, M. (1999), *Subsampling*, Springer Science & Business Media.

Quenouille, M. H. (1956), 'Notes on bias in estimation', *Biometrika* **43**(3/4), 353–360.

Quiroz, M., Kohn, R. and Dang, K.-D. (2018), 'Subsampling MCMC—an introduction for the survey statistician', *Sankhya: The Indian Journal of Statistics* **80-A**, S33–S69.

Ranganath, R., Gerrish, S. and Blei, D. (2014), Black Box Variational Inference, *in* 'International Conference on Artificial Intelligence and Statistics'.

Robert, C. and Casella, G. (2004), *Monte Carlo Statistical Methods*, 2nd edn, Springer.

Robert, C. and Casella, G. (2011), 'A short history of Markov Chain Monte Carlo: subjective recollections from incomplete data', *Statistical Science* **26**(1), 102–115.

Rohe, K., Chatterjee, S. and Yu, B. (2011), 'Spectral clustering and the high-dimensional stochastic blockmodel', *The Annals of Statistics* **39**(4), 1878–1915.

Rombach, P., Porter, M. A., Fowler, J. H. and Mucha, P. J. (2017), 'Core-periphery structure in networks (revisited)', *SIAM Review* **59**(3), 619–646.

Snijders, T. A. and Nowicki, K. (1997), 'Estimation and prediction for stochastic blockmodels for graphs with latent block structure', *Journal of classification* **14**(1), 75–100.

Todeschini, A., Miscouridou, X. and Caron, F. (2020), 'Exchangeable random measures for sparse and modular graphs with overlapping communities', *Journal of the Royal Statistical Society B* .

Tokdar, S. T. and Kass, R. E. (2010), 'Importance sampling: a review', *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(1), 54–60.

Tukey, J. (1958), 'Bias and confidence in not quite large samples', *Ann. Math. Statist.* **29**, 614.

Watanabe, S. (1968), 'A limit theorem of branching processes and continuous state branching processes', *Journal of Mathematics of Kyoto University* **8**(1), 141–167.

Welling, M. and Teh, Y. W. (2011), Bayesian learning via stochastic gradient Langevin dynamics, *in* 'International Conference on Machine Learning'.

Wolfe, P. and Choi, D. (2014), 'Co-clustering separately exchangeable network data', *Annals of Statistics* **42**(1), 29–63.

Wolfe, P. J. and Olhede, S. C. (2013), 'Nonparametric graphon estimation', *arXiv preprint arXiv:1309.5936* .

Zhang, J. Y., Khanna, R., Kyrillidis, A. and Koyejo, O. (2019), 'Learning sparse distributions using iterative hard thresholding', *arXiv preprint arXiv:1910.13389* .

Zhang, J. Y., Khanna, R., Kyrillidis, A. and Koyejo, O. (2020), 'Bayesian coresets: An optimization perspective', *arXiv preprint arXiv:2007.00715* .

Zhang, Y., Levina, E. and Zhu, J. (2017), 'Estimating network edge probabilities by neighbourhood smoothing', *Biometrika* **104**(4), 771–783.

Zhao, Y., Levina, E. and Zhu, J. (2012), 'Consistency of community detection in networks under degree-corrected stochastic block models', *The Annals of Statistics* **40**(4), 2266–2292.