Bayesian Analysis (2024) **TBA**, Number TBA, pp. 1–31

Posterior Sampling From Truncated Ferguson-Klass Representation of Normalised Completely Random Measure Mixtures

Junyi Zhang* and Angelos Dassios†

Abstract. In this paper, we study the finite approximation of the completely random measure (CRM) by truncating its Ferguson-Klass representation. The approximation is obtained by keeping the N largest atom weights of the CRM unchanged and combining the smaller atom weights into a single term. We develop the simulation algorithms for the approximation and characterise its posterior distribution, for which a blocked Gibbs sampler is devised. We demonstrate the usage of the approximation in two models. The first assumes such an approximation as the mixing distribution of a Bayesian nonparametric mixture model and leads to a finite approximation to the model posterior. The second concerns the finite approximation to the Caron-Fox model. Examples and numerical implementations are given based on the gamma, stable and generalised gamma processes.

MSC2020 subject classifications: Primary 62C10; secondary 60G57.

Keywords: Bayesian nonparametric statistics, completely random measures, blocked Gibbs sampler, approximate inference, generalised gamma process.

1 Introduction

Completely random measures (CRMs) (Kingman 1967) and their normalisations (known as the normalised random measures with independent increments, or NRMIs) (Regazzini et al. 2003) are a rich and flexible class of Bayesian nonparametric priors. The most fundamental nonparametric prior, namely the Dirichlet process (Ferguson 1973), is an NRMI derived by normalising the increments of a gamma process. Other celebrated examples include the normalised stable process (Kingman 1975), the normalised inverse-Gaussian (NIG) process (Lijoi et al. 2005), the normalised generalised gamma (NGG) process (Brix 1999, Lijoi et al. 2007, Barrios et al. 2013) and the beta process (Hjort 1990, Kim 1999, Thibaux and Jordan 2007, Broderick et al. 2012). As explained by James et al. (2009), the variety of nonparametric priors mainly arises from their high flexibility in density estimation and clustering problems, with the aim of overcoming some of the drawbacks of the Dirichlet process. In fact, the Dirichlet process allocates the observations into clusters with probabilities depending only on the current cluster sizes and the concentration parameter. The NIG and NGG processes generalise the Dirichlet process by adding extra parameters to the cluster allocation probabilities,

*Department of Applied Mathematics, The Hong Kong Polytechnic University, Kowloon, Hong Kong, JunyZhang@polyu.edu.hk

†Department of Statistics, London School of Economics, London WC2A 2AE, UK, A.Dassios@lse.ac.uk

© 2024 International Society for Bayesian Analysis https://doi.org/10.1214/24-BA1421

thus having further degrees of freedom compared to the Dirichlet process, and their posterior analysis relies on the properties of the underlying CRMs. We refer to Phadia (2016) and Ghosal and van der Vaart (2017) for a comprehensive review of the different CRMs and their properties. In addition, we remark that the CRMs are also related to the Gibbs-type priors, a generalisation of the Dirichlet process. A systematic and unified treatment of the Gibbs–type priors can be found in De Blasi et al. (2015).

Since the distribution of the nonparametric prior involves an infinite number of variables, an explicit representation of the prior is not available in practice, as we cannot simulate or store infinitely many random variables. To facilitate the posterior inference, the existing literature has introduced two major approaches: the marginal and conditional schemes. The marginal scheme integrates out the infinite number of latent variables and leads to a Pólya urn scheme for the predictive distribution. See Ishwaran and James (2001) and Favaro and Teh (2013) for further discussion. For the Dirichlet process mixture model, we refer the reader to Algorithm 8 of Neal (2000), which is the standard marginal posterior inference scheme. The conditional scheme, on the other hand, focuses on the finite representation of the nonparametric prior. Ishwaran and James (2001) and Ishwaran and James (2002) truncated the stick-breaking representation of the Dirichlet and Pitman-Yor processes at a deterministic level and analysed the truncation error. Walker (2007) and Kalli et al. (2011) proposed a slice sampler that avoids the user-specified truncation level and leads to an exact sampling scheme. Muliere and Tardella (1998) proposed a random stopping rule for the Dirichlet process, which allows a priori control of the truncation error. Arbel et al. (2019) extended this method to the Pitman-Yor process. Lee et al. (2016) used the finite-dimensional BFRY process (Bertoin et al. 2006) to approximate the generalised gamma process. Argiento et al. (2016b) developed the ϵ-approximation for the general CRMs by discarding the atom weights smaller than a threshold, and an implementation of this method for the NGG process can be found in Argiento et al. (2016a). Lee et al. (2023) proposed a general and unified framework to derive both series representation and finite-dimensional approximation of the CRMs. Their method includes several well-known representations and approximations of the CRMs, including the Ferguson-Klass representation and the BFRY process approximation, as special cases.

In this work, we introduce a novel finite approximation to the general CRMs and their normalisations. Our approximation is obtained by (i) sorting the atom weights of the CRM in descending order and including the N atoms with the largest weights in the approximation; (ii) combining the infinite number of atoms with smaller weights into a single term and retaining it in the approximation. Thus, our method produces an $(N+1)$ dimensional approximation for the CRM. Although the idea of ranked atom weights has been considered in Ferguson and Klass (1972), Walker and Damien (2000), Griffin (2016), Arbel and Prünster (2017) and Lee et al. (2023), our method differs from the existing works in the way that the smaller atom weights, which are usually ignored, are retained in the finite approximation. As a result, our approximation has the same total mass as the CRM. On the other hand, Muliere and Tardella (1998) and Arbel et al. (2019) truncated the stick-breaking representation of the Dirichlet and Pitman-Yor processes and assigned the residual mass to a single term. However, this approach relies on the availability of the stick-breaking representation, which is still an open problem

for general NRMIs. Also, in the stick-breaking representation, the atom weights are only decreasing in the mean. Hence, our approximation has the key advantages of preserving the total mass of the CRM and producing almost surely decreasing atom weights. Such an approximation method was initially introduced by Zhang and Dassios (2023), where the focus was solely given to the Dirichlet process. In the current work, we extend this method to the general CRMs. To this end, we will make a comprehensive study about the simulation algorithms of the finite approximation of CRMs. Also, the posterior inference scheme developed by Zhang and Dassios (2023) involves the derivative of the density of a truncated gamma process, which is overcomplicated and unnecessary. The current work will provide a simplified inference scheme that is easier to derive and implement.

We will illustrate the usage of our approximation in the Bayesian nonparametric mixture model and the Caron-Fox model. The analysis of the mixture model is the most fundamental problem in Bayesian nonparametric statistics. In this work, we develop a new posterior inference scheme based on the truncated Ferguson-Klass representation of the NRMI prior. We will illustrate the effectiveness of this method in the normal-normal/inverse gamma model based on the NGG process mixing distribution and compare it with the BNPdensity package developed by Arbel et al. (2021).

The Caron-Fox model is a celebrated innovation that introduces Bayesian nonparametric methods to statistical network modelling. It assumes that the sociabilities of the nodes in the network are generated by the atom weights of a CRM. Due to the CRM prior, the model's posterior involves infinite variables. Caron and Fox (2017) suggested dividing the nodes into two types, namely the nodes with at least one edge (the active nodes) and the nodes without any edge (the silent nodes). Since the number of edges is finite in the observation, the number of active nodes must also be finite. Then, they developed an MCMC algorithm that estimates the sociabilities of the active node and the total sociability of the silent nodes. Alternatively, Li and Campbell (2021) truncated the sequential representation of the CRM at a fixed level and derived the upper and lower bounds of the probability that all the active nodes are in the truncation region. They also developed an adaptive Metropolis-Hastings sampler for posterior inference. In addition, for the network model based on the generalised gamma process prior, Naik et al. (2022) approximated the prior with the approximations proposed by Lee et al. (2023) and designed an MCMC sampler for posterior inference.

In this paper, we propose a new posterior approximation method for the Caron-Fox model. The proposed approach is motivated by the fact that there could be a significant difference between the number of edges possessed by the active nodes, and we are more interested in the nodes with more edges. In a social network, for example, a few popular users usually enjoy a large number of followers, while the majority of users have very few followers. In this case, we are more interested in the most popular users, and the users with very few followers are usually less relevant to practical applications such as the social network recommendation system. When the data set contains a huge number of nodes, we want to focus on the N nodes with the highest sociabilities and combine the other nodes, either with fewer edges or without any edge, into a single term. This can be achieved by replacing the CRM prior with the truncated Ferguson-Klass representation developed in this work.

The contribution of this paper includes the following three aspects: (i) a new finite approximation for the CRMs which preserves the total mass of the CRM and produces almost surely decreasing atom weights, (ii) a new posterior inference scheme which significantly simplifies the algorithm in Zhang and Dassios (2023), and (iii) a new truncated simulation and approximate inference scheme for the Caron-Fox model.

The rest of the paper is organised as follows. Section 2 reviews the basic properties of the CRMs and NRMIs. We also briefly introduce the Bayesian nonparametric mixture model and the Caron-Fox model in this section. Section 3 constructs the truncated Ferguson-Klass representation of the CRMs and develops the simulation algorithms for the approximation. Section 4 devises a blocked Gibbs sampler for the posterior of the approximation process. Section 5 presents examples of the proposed approximation for several well-known CRMs. Section 6 contains the numerical implementations. Section 7 discusses the perspective of the work.

2 Preliminaries

In this section, we briefly review the characterisation of a homogeneous pure-atomic completely random measure via the increments of a subordinator. Consider a Lévy process τ on $\mathbb{R}_+ := [0, \infty)$ with the Lévy-Khintchine representation

$$\mathbb{E}(e^{-\beta\tau}) = \exp\left(-\int_0^\infty (1 - e^{-\beta w})\rho(dw)\right),$$

where $\rho(dw)$ is a Lévy measure satisfying

$$\int_0^\infty \rho(dw) = \infty \text{ and } \int_0^\infty \min(1, w)\rho(dw) < \infty.$$

The Lévy process will generate a countable infinity of jumps $\{\tilde{J}_i\}_{i\geq 1}$, for $\tilde{J}_i \in \mathbb{R}_+$, having an almost surely finite sum $\tau = \sum_{i=1}^\infty \tilde{J}_i < \infty$. Thus, τ is known as a pure-jump infinite-activity Lévy process, or a subordinator. Throughout this paper, we will use the tilde notation to emphasise that the sequence of jumps $\{\tilde{J}_i\}_{i\geq 1}$ is presented according to the time of the jumps without any reordering.

In a Bayesian nonparametric model, each jump $\tilde{J}_i$ represents the possibility or frequency of the i-th parameter, denoted by K_i. We assume that the sequence of random variables $\{K_i\}_{i\geq 1}$ are i.i.d. with some base distribution G_0 on the support $\mathcal{S}$, which is independent of the jumps. Then, we can construct a CRM which assigns the frequency $\tilde{J}_i$ to the parameter K_i, for $i = 1, 2, \ldots$. The CRM has the format

$$\tilde{\Theta}_\infty(.) := \sum_{i=1}^\infty \tilde{J}_i \delta_{K_i}(.) \sim \text{CRM}(\rho),$$

where $\delta_{K_i}(.)$ denotes a point mass at K_i. We refer to $\tilde{\Theta}_\infty$ as a CRM derived from the Lévy measure $\rho(dw)$, abbreviated by $\tilde{\Theta}_\infty \sim \text{CRM}(\rho)$. The base distribution G_0 is left implicit in the definition as it has no effect on our analysis. The CRM itself could serve

as the prior of a Bayesian nonparametric model, and a likelihood process can be defined based on it. In the current research, however, we are more interested in the normalised CRM. Since the total mass of the CRM is given by $\tilde{\Theta}_\infty(\mathcal{S}) = \sum_{i=1}^{\infty} \tilde{J}_i = \tau$, a random probability measure can be defined through the normalisation of a CRM:

$$\tilde{\mathcal{P}}_\infty(.) := \frac{\tilde{\Theta}_\infty(.)}{\tilde{\Theta}_\infty(\mathcal{S})} = \sum_{i=1}^{\infty} \tilde{p}_i \delta_{K_i}(.) = \sum_{i=1}^{\infty} \frac{\tilde{J}_i}{\tau} \delta_{K_i}(.) \sim \text{NRMI}(\rho).$$

We refer to $\tilde{\mathcal{P}}_\infty$ as a NRMI (Regazzini et al. 2003, James et al. 2009) derived from the Lévy measure $\rho(dw)$, abbreviated by $\tilde{\mathcal{P}}_\infty \sim \text{NRMI}(\rho)$. The tilde notation has the same meaning as before, i.e., the sequence of probability masses $\{\tilde{p}_i\}_{i\geq 1}$ is presented without any reordering. Well-known nonparametric priors derived from the NRMI include the Dirichlet, NIG and NGG processes.

Using the NRMI as the mixing distribution, we consider the following Bayesian nonparametric mixture model:

$$X_i \mid Y_i \sim \mathcal{K}(X_i \mid Y_i), \;\; Y_i \mid \tilde{\mathcal{P}}_\infty \overset{iid}{\sim} \tilde{\mathcal{P}}_\infty, \;\; \tilde{\mathcal{P}}_\infty \sim \text{NRMI}(\rho), \;\; i = 1, \ldots, n, \tag{2.1}$$

where $X_1, \ldots, X_n$ represent the observations, they are assumed to be independent conditionally on the latent variables Y_i. The mixture kernel $\mathcal{K}(X_i \mid Y_i)$ denotes the conditional distribution of X_i given Y_i. The latent variables Y_i are i.i.d. according to $\tilde{\mathcal{P}}_\infty$, and the mixing distribution $\tilde{\mathcal{P}}_\infty$ is a sample of NRMI(ρ). In the successive sections, we will propose a finite approximation to the NRMI(ρ) prior and develop a posterior inference scheme for the mixture model based on it.

Apart from serving as the mixing distribution, the application of CRMs has also been found in statistical network modelling. Consider a Caron-Fox model Z in terms of

$$Z = \sum_{j=1}^{n} \delta_{\{s_j, r_j\}}, \;\; s_j, r_j \overset{iid}{\sim} \frac{\tilde{\Theta}_\infty(.)}{\tilde{\Theta}_\infty(\mathcal{S})} = \sum_{i=1}^{\infty} \frac{\tilde{J}_i}{\tau} \delta_{K_i}(.), \;\; n \sim \text{Pois}(\tau^2), \;\; \tilde{\Theta}_\infty \sim \text{CRM}(\rho), \tag{2.2}$$

where each pair of s_j and r_j stands for the indices of the origin and destination nodes of the j-th edge, for $j = 1, \ldots, n$. They are chosen independently according to the normalised random measure $\tilde{\Theta}_\infty(.)/\tilde{\Theta}_\infty(\mathcal{S})$. The latent variables $\{\tilde{J}_i\}_{i\geq 1}$ denote the sociabilities of the nodes. A node with a higher sociability is more likely to send or receive an edge. The total number of edges in the network is assumed to be a Poisson random number with mean $\tau^2 = (\tilde{\Theta}_\infty(\mathcal{S}))^2$, meaning that more edges are expected if the nodes have higher sociabilities. Assume that the observation of model (2.2) contains N_a active nodes, we will use the notation $\tilde{n}_{ij}$ for the number of edges from the i-th active node to the j-th active node, for $1 \leq i, j \leq N_a$. We also denote by $\tilde{m}_i := \sum_{j=1}^{N_a} (\tilde{n}_{ij} + \tilde{n}_{ji}) > 0$ the total number of edges sent or received by the i-th active node, for $i = 1, \ldots, N_a$.

If we remove the Poisson assumption on the total number of edges and use a fixed n instead, the actual sociabilities $\{\tilde{J}_i\}_{i\geq 1}$ have no effect on the distribution of the edges, and only their proportions are relevant. Then we can replace the CRM(ρ) prior by its normalisation NRMI(ρ). To this end, Williamson (2016) introduced a network model based on the Dirichlet process prior and used it to study the link prediction problem.

3 Finite approximation of CRMs

The goal of this section is the development of the truncated Ferguson-Klass representation of the CRMs and their normalisations. Since the random variables $\{K_i\}_{i\geq 1}$ are i.i.d., the CRM $\tilde{\Theta}_\infty(.) = \sum_{i=1}^{\infty} \tilde{J}_i \delta_{K_i}(.)$ is identical in distribution to a reordering of its components. In particular, let $J_1 > J_2 > \dots$ be the ranked values of $\{\tilde{J}_i\}_{i\geq 1}$, then the Ferguson-Klass representation of a CRM is given by

$$\Theta_\infty(.) := \sum_{i=1}^{\infty} J_i \delta_{K_i}(.).$$

The ranked jumps can be derived from the inverse Lévy measure $J_i := \rho^{\leftarrow}(\Gamma_i)$, where $\rho^{\leftarrow}(w) := \inf\{x \mid \rho(x,\infty) \leq w\}$, $\rho(x,\infty) := \int_x^\infty \rho(dw)$ denotes the tail distribution of the Lévy measure, $\Gamma_i := \sum_{j=1}^{i} E_j$, and $E_j \sim \text{Exp}(1)$ are i.i.d. exponential random variables with mean 1. Although $\tilde{\Theta}_\infty$ and Θ_∞ are identical in distribution, they lead to different truncation methods. We consider the following finite approximation to Θ_∞:

$$\Theta_N(.) := \sum_{i=1}^{N} J_i \delta_{K_i}(.) + {}^{(N)}\tau \delta_{K_0}(.) \sim \text{N-CRM}(\rho),$$

where K_0 is an independent sample of G_0, and ${}^{(N)}\tau := \sum_{i=N+1}^{\infty} J_i$ stands for the N-trimmed subordinator (Ipsen and Maller 2017) derived from τ, i.e., the process obtained by removing the N largest jumps from τ. We refer to Θ_N as the truncated Ferguson-Klass representation of $\tilde{\Theta}_\infty$, abbreviated by N-CRM(ρ). It follows immediately that $\Theta_N \to \tilde{\Theta}_\infty$ in distribution almost surely as $N \to \infty$. Based on a decreasing sequence of atom weights, the truncated Ferguson-Klass representation has the lowest approximating error compared to other truncated sequential representations of the CRMs. See Campbell et al. (2019) for more details.

Since $\tilde{\Theta}_\infty(\mathcal{S}) = \Theta_N(\mathcal{S}) = \tau$, our approximation has the same total mass as the CRM. Therefore, we can easily construct a finite approximation to NRMI(ρ). Let $p_1 > p_2 > \dots$ be the ranked values of the probability masses $\{\tilde{p}_i\}_{i\geq 1}$. The truncated Ferguson-Klass representation of $\tilde{\mathcal{P}}_\infty$ is given by

$$\mathcal{P}_N(.) := \frac{\Theta_N(.)}{\Theta_N(\mathcal{S})} = \sum_{i=1}^{N} p_i \delta_{K_i}(.) + e_N \delta_{K_0}(.) \sim \text{N-NRMI}(\rho), \tag{3.1}$$

where the N largest probability masses have the representation $p_i = J_i/\tau$, for $i = 1, \dots, N$, and the sum of smaller probability masses has the representation $e_N := \sum_{i=N+1}^{\infty} p_i = \sum_{i=N+1}^{\infty} J_i/\tau$. We refer to $\mathcal{P}_N$ as the truncated Ferguson-Klass representation of $\tilde{\mathcal{P}}_\infty$, abbreviated by N-NRMI($\rho$). By definition, $\mathcal{P}_N \to \tilde{\mathcal{P}}_\infty$ in distribution almost surely as $N \to \infty$. Thus, $\mathcal{P}_N$ provides a finite approximation to the distribution of $\tilde{\mathcal{P}}_\infty$, and the approximation error is revealed by the tail probability e_N.

From the discussion above, it is clear that the construction of N-CRM(ρ) and N-NRMI(ρ) depends on the N largest jumps and the sum of smaller jumps, namely $(J_1, \dots, J_N, {}^{(N)}\tau)$, of the subordinator τ. To derive their joint density, we first prepare a lemma concerning the distribution of a N-trimmed subordinator.

Lemma 3.1. *Let $\{J_i\}_{i=1}^{\infty}$ be the ranked jumps of a subordinator τ with Lévy measure $\rho(dw)$ such that $J_1 > J_2 > \dots$ and $\tau = \sum_{i=1}^{\infty} J_i$. Denote by ${}^{(N)}\tau := \sum_{i=N+1}^{\infty} J_i$ the N-trimmed subordinator derived from τ. The conditional density of ${}^{(N)}\tau$, given $J_1, \dots, J_N$, is*

$$f_{\rho,J_N}(z) = \sum_{i=0}^{n-1} \frac{(-t)^i}{i!} L_i(z),$$

where $L_i(z)$ is defined recursively as follows:

$$L_0(z) = f_\rho(z) \exp\left(\int_{J_N}^{\infty} \rho(dw)\right), \quad \text{for } z > 0,$$

$$L_{i+1}(z) = \int_{J_N}^{z-iJ_N} L_i(z-w)\rho(dw), \quad \text{for } z > (i+1)J_N,$$

and $f_\rho(z)$ stands for the density of τ.

From the basic properties of the Poisson random measure (see, for example, Section 2.2 of Kyprianou 2006), we obtain the joint density of $(J_1, \dots, J_N, {}^{(N)}\tau)$ as follows,

$$\begin{aligned} &\mathbb{P}(J_1 \in dx_1, \dots, J_N \in dx_N, {}^{(N)}\tau \in dy) \\ &= e^{-\rho(x_N,\infty)} \rho(x_1) \dots \rho(x_N) f_{\rho,x_N}(y) dy dx_N \dots dx_1, \end{aligned} \tag{3.2}$$

where $x_1 > x_2 > \dots > x_N > 0$ and $y \in (0, \infty)$. However, in simulation and posterior inference problems, it is more convenient to work with the ratio between two consecutive jumps. To this end, we provide an alternative expression for the joint density in the following theorem.

Theorem 3.2. *Under the settings of Lemma 3.1, denote by $R_k := J_{k+1}/J_k$ the ratio between the $(k+1)$-th and k-th largest jumps of the subordinator τ, then the joint density of $(R_1, \dots, R_{N-1}, J_N, {}^{(N)}\tau)$ is*

$$\begin{aligned} &\mathbb{P}(R_1 \in dr_1, \dots, R_{N-1} \in dr_{N-1}, J_N \in dz, {}^{(N)}\tau \in dy) \\ &= e^{-\rho(z,\infty)} \rho(z) \Pi^{-1} \left(\prod_{i=1}^{N-1} (z\Pi^{-1} r_1 \dots r_{i-1}) \rho(z\Pi^{-1} r_1 \dots r_{i-1}) \right) \\ &\quad \times f_{\rho,z}(y) dy dz dr_{N-1} \dots dr_1, \end{aligned} \tag{3.3}$$

where $\Pi := r_1 \dots r_{N-1}$, $r_i \in (0,1)$, for $i = 1, \dots, N-1$, $z \in (0, \infty)$, $y \in (0, \infty)$, and $f_{\rho,z}(y)$ is given in Lemma 3.1 with $J_N = z$.

Next, we introduce four different simulation algorithms for N-CRM(ρ): the ILM algorithm, the rejection algorithm, the AR algorithm and the MCMC algorithm. These algorithms differ from each other in the simulation method of the N largest jumps $J_1, \dots, J_N$, and a suitable choice of the algorithm is critical for the efficient simulation of the jumps. On the other hand, the simulation algorithm for ${}^{(N)}\tau$ varies for different Lévy measures. Thus, we use a generic description here and illustrate the details for some

celebrated nonparametric priors in Section 5. We will also compare the performance of the different algorithms in Section 6 to demonstrate the importance of the variety of the simulation algorithms.

To sample from the N largest jumps, it is straightforward to use the inverse Lévy measure (ILM) algorithm (Wolpert and Ickstadt 1998) given as follows.

Algorithm 3.3 (ILM algorithm)**.** *Simulation algorithm for* $(J_1, \ldots, J_N, {}^{(N)}\tau)$.

1. *Set* $\Gamma_0 \leftarrow 0$.
2. *For* $i = 1, \ldots, N$, *set* $\Gamma_i \leftarrow \Gamma_{i-1} + Exp(1)$, *then solve the equation* $\Gamma_i = \int_x^\infty \rho(dw)$ *and set* $J_i \leftarrow x$.
3. *Simulate a subordinator* ${}^{(N)}\tau$ *with the Lévy measure* $\rho(dw)\mathbb{1}_{(0,J_N)}(w)$.

The ILM algorithm is preferable when the tail distribution of the Lévy measure admits an explicit inversion. However, apart from a few special cases, for example, the stable process, the tail distribution has no close form, and we must solve the equations in Step 2 numerically. The integral in the equation might cause a computational problem, but we can use special functions to improve the performance of the algorithm. For example, the Lévy measure of the gamma process has the tail distribution $\int_x^\infty w^{-1}e^{-w}dw = E_1(x)$, where $E_1(\cdot)$ denotes the exponential integral. The Lévy measure of the generalised gamma process has the tail distribution $\int_x^\infty w^{-\alpha}e^{-\mu w}dw = \mu^{\alpha-1}\Gamma(1-\alpha, x\mu)$, where $\Gamma(\cdot,\cdot)$ denotes the upper incomplete gamma function. Efficient computation methods for these functions have been well studied in the existing literature, and the packages are available in various programming languages.

Alternatively, we can use the rejection algorithm to sample from $J_1, \ldots, J_N$. The algorithm is based on the rejection representation (Rosiński 2001) of the CRMs. To implement this algorithm, we need to find an envelope Lévy measure $\rho_0(dw)$, such that $d\rho/d\rho_0 \leq 1$ uniformly and the ILM algorithm is easy to use for ρ_0. The details of the algorithm are given as follows.

Algorithm 3.4 (Rejection algorithm)**.** *Simulation algorithm for* $(J_1, \ldots, J_N, {}^{(N)}\tau)$.

1. *Set* $k \leftarrow 1$, $i \leftarrow 1$, $\Gamma_0 \leftarrow 0$.
2. *While* $k \leq N$
   (a) *Set* $\Gamma_i \leftarrow \Gamma_{i-1} + Exp(1)$, *then solve the equation* $\Gamma_i = \int_x^\infty \rho_0(dw)$ *and set* $J_i^0 \leftarrow x$.
   (b) *Set* $U_i \leftarrow U(0,1)$. *If* $U_i \leq \rho(J_i^0)/\rho_0(J_i^0)$, *set* $J_k \leftarrow J_i^0$ *and* $k \leftarrow k+1$.
   (c) *Set* $i \leftarrow i+1$.
3. *Simulate a subordinator* ${}^{(N)}\tau$ *with the Lévy measure* $\rho(dw)\mathbb{1}_{(0,J_N)}(w)$.

The expected number of rejections in Step 2 can be found in Campbell et al. (2019). Note that the rejection algorithm is particularly useful for the Dirichlet process. See Section 5 for more details.

Another approach is to use the acceptance-rejection (AR) method to sample from $J_1, \ldots, J_N$. This method hinges on the availability of the envelopes for the target densities in Step 1 and 2. The details of the algorithm are given as follows.

Algorithm 3.5 (AR algorithm)**.** *Simulation algorithm for* $(J_1, \ldots, J_N, {}^{(N)}\tau)$.

1. *Sample from the target density*

$$\exp\left(-\int_{x_1}^{\infty} \rho(dw)\right) \rho(x_1), \quad \text{for } x_1 \in (0, \infty),$$

using the acceptance-rejection method, set the outcome as J_1.

2. *For* $k = 2, \ldots, N$, *sample from the target density*

$$\exp\left(-\int_{x_k}^{J_{k-1}} \rho(dw)\right) \rho(x_k), \quad \text{for } x_k \in (0, J_{k-1}),$$

using the acceptance-rejection method, set the outcome as J_k.

3. *Simulate a subordinator* ${}^{(N)}\tau$ *with the Lévy measure* $\rho(dw)\mathbb{1}_{(0,J_N)}(w)$.

In addition, we can use the Markov chain Monte Carlo (MCMC) method to draw from the joint density (3.3). This algorithm is simply described as running Algorithm 4.2, which will be explained later, iteratively with $n = 0$. We refer to this algorithm as the MCMC algorithm.

We remark that the outcomes of all four algorithms above share the same distribution. In fact, Rosiński (2001) has shown the validity of the rejection representation of the CRMs. Using the ILM algorithm as the proposal in Step 2 of Algorithm 3.4, we are actually sampling from the N largest atom weights of the rejection representation, and thus that of the CRM. On the other hand, the AR algorithm is simply a version of the ILM algorithm when the inverse Lévy measure is not available in close form. The effectiveness of the MCMC algorithm is clear as it draws from the target density (3.3) directly. We remark that it is straightforward to set $\tau \leftarrow J_1 + \cdots + J_N + {}^{(N)}\tau$ and $(p_1, \ldots, p_N, e_N) \leftarrow (J_1/\tau, \ldots, J_N/\tau, {}^{(N)}\tau/\tau)$ in the last step of these algorithms, such that the output becomes a sample of the probability masses of N-NRMI(ρ).

To close this section, we discuss the approximation error of N-CRM(ρ). From the definition, it is clear that the approximation error is revealed by ${}^{(N)}\tau$. To derive the distribution of ${}^{(N)}\tau$, it is theoretically possible to integrate out $R_1, \ldots, R_{N-1}, J_N$ from the joint density (3.3), that is,

$$\begin{aligned}&\mathbb{P}({}^{(N)}\tau \in dy)\\ &= \int_{0<r_1,\ldots,r_{N-1}<1,z>0} \mathbb{P}(R_1 \in dr_1, \ldots, R_{N-1} \in dr_{N-1}, J_N \in dz, {}^{(N)}\tau \in dy),\end{aligned}$$

for $y > 0$. Similarly, the approximation error of N-NRMI(ρ) is given by e_N. We could express e_N in terms of

$$\begin{aligned}&\{e_N = y \mid R_1, \ldots, R_{N-1}, J_N\}\\ &= \left\{ {}^{(N)}\tau = \frac{y}{1-y} J_N (R_1^{-1} \ldots R_{N-1}^{-1} + \cdots + R_{N-1}^{-1} + 1) \mid R_1, \ldots, R_{N-1}, J_N \right\},\end{aligned}$$

and use the joint distribution of $R_1, \ldots, R_{N-1}, J_N$ to integrate out the conditionals,

$$\begin{aligned}\mathbb{P}(e_N \in dy) = \int_{0<r_1,\ldots,r_{N-1}<1,z>0} & f_{\rho,z}\left(\frac{y}{1-y} z(r_1^{-1} \ldots r_{N-1}^{-1} + \cdots + r_{N-1}^{-1} + 1)\right)\\ &\times \mathbb{P}(R_1 \in dr_1, \ldots, R_{N-1} \in dr_{N-1}, J_N \in dz),\end{aligned}$$

for $y \in (0, 1)$. However, for many heavily used Lévy measures, these integrals would be too complicated to calculate in close form. For this reason, we suggest using the simulation algorithms introduced in this section to estimate $\mathbb{E}({}^{(N)}\tau)$ and $\mathbb{E}(e_N)$ numerically. An numerical implementation of this method will be given in Section 6.

4 Posterior inference

In this section, we study the posterior inference problems of the models based on the N-CRM(ρ) and N-NRMI(ρ) priors. For the Bayesian nonparametric mixture model (2.1), we replace the NRMI prior by the finite approximation N-NRMI(ρ) and consider the following parametric model,

$$X_i \mid Y_i \sim \mathcal{K}(X_i \mid Y_i), \;\; Y_i \mid \mathcal{P}_N \overset{iid}{\sim} \mathcal{P}_N, \;\; \mathcal{P}_N \sim \text{N-NRMI}(\rho), \;\; i = 1, \ldots, n. \tag{4.1}$$

Recall that Lemma 5.2 of Campbell et al. (2019) (see also Campbell 2016) showed that

$$0.5||p_{n,\infty} - p_{n,N}||_1 \leq 1 - \mathbb{P}(X_{1:n} \subseteq \text{support}(\mathcal{P}_N)),$$

where $p_{n,\infty}$ and $p_{n,N}$ are the marginal densities of the observations $X_1, \ldots, X_n$ from model (2.1) and (4.1), respectively. Since the supports of $\tilde{\mathcal{P}}_\infty$ and $\mathcal{P}_N$ overlap only on the first N terms of $\mathcal{P}_N$, i.e., $K_1, \ldots K_N$, Jensen's inequality implies that

$$\begin{aligned}\mathbb{P}(X_{1:n} \subseteq \text{support}(\mathcal{P}_N)) = \mathbb{E}\left[\left(\frac{J_1 + \cdots + J_N}{\sum_{i=1}^\infty J_i}\right)^n\right] &\geq \mathbb{E}\left(\frac{J_1 + \cdots + J_N}{\sum_{i=1}^\infty J_i}\right)^n\\ &= \mathbb{P}(X_1 \in \text{support}(\mathcal{P}_N))^n = (1 - \mathbb{E}(e_N))^n.\end{aligned}$$

It follows that the marginal densities $p_{n,\infty}$ and $p_{n,N}$ would become closer if there are more overlaps between the supports of $\tilde{\mathcal{P}}_\infty$ and $\mathcal{P}_N$. When $N \to \infty$, $\mathbb{P}(X_{1:n} \subseteq \text{support}(\mathcal{P}_N)) \to 1$, and the two marginal densities become identical. Thus, the posterior of model (4.1) provides a finite approximation to that of model (2.1). The convergence rate between the two marginal densities relies on $\mathbb{E}(e_N)$, in other words, the expectation of the approximation error of the prior. See Theorem 5.3 of Campbell et al. (2019) for a further derivation of the error bound.

To sample from the posterior of model (4.1), we first consider a special case where the mixture kernel $\mathcal{K}(X_i \mid Y_i)$ is a point mass at Y_i. In this case, model (2.1) reduces to a hierarchical model in terms of

$$X_i \mid \tilde{\mathcal{P}}_\infty \overset{iid}{\sim} \tilde{\mathcal{P}}_\infty, \;\; \tilde{\mathcal{P}}_\infty \sim \text{NRMI}(\rho), \;\; i = 1, \ldots, n. \tag{4.2}$$

Using the same argument as before, we replace the NRMI prior by the finite approximation N-NRMI(ρ) and consider the following finite-dimensional hierarchical model,

$$X_i \mid \mathcal{P}_N \overset{iid}{\sim} \mathcal{P}_N, \;\; \mathcal{P}_N \sim \text{N-NRMI}(\rho), \;\; i = 1, \ldots, n. \tag{4.3}$$

Next, we devise a blocked Gibbs sampler for the posterior of model (4.3). The sampler will be used in the posterior inference of the Bayesian nonparametric mixture model and the Caron-Fox model.

4.1 Blocked Gibbs sampler

We first characterise the posterior of model (4.3). Denote by $\Sigma := 1 + R_1 + \cdots + R_1 \ldots R_{N-1}$ and rewrite the representations of the probability masses in (3.1) as

$$p_i = \frac{J_N \Pi^{-1} R_1 \ldots R_{i-1}}{J_N \Pi^{-1} \Sigma + {}^{(N)}\tau}, \;\; \text{for } i = 1, \ldots, N, \;\; e_N = \frac{{}^{(N)}\tau}{J_N \Pi^{-1} \Sigma + {}^{(N)}\tau}. \tag{4.4}$$

Let $X_1, \ldots, X_n$ be the observations of model (4.3), and let $n_i = \text{card}\{X_j = K_i\}$ be the number of observations which equal K_i, for $i = 0, 1 \ldots, N$, then $n = \sum_{i=0}^{N} n_i$ is the total number of observations. Conditioning on $p_1, \ldots, p_N, e_N$, the likelihood of $\mathcal{P}_N$ follows the multinomial distribution Multi$(n_1, \ldots, n_N, n_0; p_1, \ldots, p_N, e_N)$. Thus, the posterior of model (4.3) is proportional to

$$\begin{aligned}
&\mathbb{P}(R_1 \in dr_1, \ldots, R_{N-1} \in dr_{N-1}, J_N \in dz, {}^{(N)}\tau \in dy \mid \mathbf{K}) \\
&\propto z^{n_1 + \cdots + n_N} r_1^{-n_1} \ldots r_{N-1}^{-(n_1 + \cdots + n_{N-1})} y^{n_0} (z\Pi^{-1}\Sigma + y)^{-n} e^{-\rho(z,\infty)} \rho(z) \Pi^{-1} \\
&\quad \times \left(\prod_{i=1}^{N-1} (z\Pi^{-1} r_1 \ldots r_{i-1}) \rho(z\Pi^{-1} r_1 \ldots r_{i-1}) \right) f_{\rho,z}(y) dy dz dr_{N-1} \ldots dr_1,
\end{aligned} \tag{4.5}$$

where $\mathbf{K} := (n_0, n_1, \ldots, n_N)$. To sample from (4.5), we develop a blocked Gibbs sampler to draw from $\mathbb{P}(R_1, \ldots, R_{N-1} \mid J_N, {}^{(N)}\tau, \mathbf{K})$ and $\mathbb{P}(J_N, {}^{(N)}\tau \mid R_1, \ldots, R_{N-1}, \mathbf{K})$ iteratively. For the former distribution, we apply the Hamiltonian Monte Carlo algorithm (HMC, see Neal 2011) to sample from

$$\begin{aligned}
&\mathbb{P}(R_1 \in dr_1, \ldots, R_{N-1} \in dr_{N-1} \mid J_N = z, {}^{(N)}\tau = y, \mathbf{K}) \\
&\propto r_1^{-n_1} \ldots r_{N-1}^{-(n_1 + \cdots + n_{N-1})} (z\Pi^{-1}\Sigma + y)^{-n} \Pi^{-1} \\
&\quad \times \left(\prod_{i=1}^{N-1} (\Pi^{-1} r_1 \ldots r_{i-1}) \rho(z\Pi^{-1} r_1 \ldots r_{i-1}) \right) dr_{N-1} \ldots dr_1.
\end{aligned}$$

The HMC algorithm is based on the gradient of the log-posterior which, after making the change of variables $\mathcal{R}_i := \tan(\pi(r_i - 0.5))$, for $i = 1, \ldots, N-1$, is denoted by $D(\mathcal{R}_1, \ldots, \mathcal{R}_{N-1}) := \nabla \log(\mathbb{P}(\mathcal{R}_1, \ldots, \mathcal{R}_{N-1} \mid z, y, \mathbf{K}))$. The full steps of the HMC algorithm are given as follows.

Algorithm 4.1 (HMC algorithm)**.** *Let $L \geq 1$ be the number of leapfrog steps and $\epsilon > 0$ be the step size.*

1. *Load the current values of $(r_1, \ldots, r_{N-1})$, set*
$$W^{old} \leftarrow (\tan(\pi(r_1 - 0.5)), \ldots, \tan(\pi(r_{N-1} - 0.5))).$$
2. *Sample from a normal distribution $p \leftarrow \mathcal{N}(0, I_{N-1})$, set $\tilde{p}^{(0)} \leftarrow p + (\epsilon/2)D(W^{old})$.*
3. *For $l = 1, \ldots, L-1$, set $W^{(l)} \leftarrow W^{(l-1)} + \epsilon\tilde{p}^{(l-1)}$ and $\tilde{p}^{(l)} \leftarrow \tilde{p}^{(l-1)} + \epsilon D(W^{(l)})$.*
4. *Set $W^{new} \leftarrow W^{(L-1)} + \epsilon\tilde{p}^{(L-1)}$ and $\tilde{p} \leftarrow -\{\tilde{p}^{(L-1)} + (\epsilon/2)D(W^{new})\}$, then set*
$$(\tilde{r}_1, \ldots, \tilde{r}_{N-1}) \leftarrow 0.5 + (1/\pi)\arctan(W^{new}).$$
5. *Sample from a uniform distribution $U \leftarrow U(0,1)$, if*
$$U \leq \frac{\mathbb{P}(\tilde{r}_1, \ldots, \tilde{r}_{N-1} \mid z, y, \boldsymbol{K})}{\mathbb{P}(r_1, \ldots, r_{N-1} \mid z, y, \boldsymbol{K})} \exp\left(-\frac{1}{2}\sum_{i=1}^{N-1}(\tilde{p_i}^2 - p_i^2)\right),$$
accept the candidates and output $(\tilde{r}_1, \ldots, \tilde{r}_{N-1})$; otherwise, output $(r_1, \ldots, r_{N-1})$.

Next, we use the Metropolis-Hastings algorithm to sample from the posterior of J_N and ${}^{(N)}\tau$, that is,

$$\begin{aligned}&\mathbb{P}(J_N \in dz, {}^{(N)}\tau \in dy \mid R_1 = r_1, \ldots R_{N-1} = r_{N-1}, \mathbf{K})\\ &\propto z^{n_1+\cdots+n_N} y^{n_0}(z\Pi^{-1}\Sigma + y)^{-n} e^{-\rho(z,\infty)}\rho(z)\left(\prod_{i=1}^{N-1} z\rho(z\Pi^{-1}r_1 \ldots r_{i-1})\right) f_{\rho,z}(y)dydz.\end{aligned}$$

The choice of the transition kernel depends on the specifications of the Lévy measure. If the expression of the posterior contains e^{-Cz}, where C is a number independent of z and y, then the proposal $\tilde{z} \sim \text{Ga}(n_1 + \cdots + n_N + N, C)$ might be suitable, and we can draw $\tilde{y} \sim f_{\rho,\tilde{z}}(y)$ via the corresponding simulation algorithm. If the posterior is not in a standard format, we may choose the log-normal transition kernel. In this case, the adaptive Metropolis-Hastings algorithm could be used to improve the performance of the algorithm. See Haario et al. (2001) and Griffin and Stephens (2013) for more details.

We can now formulate the blocked Gibbs sampler as follows.

Algorithm 4.2 (Blocked Gibbs sampler)**.** *Posterior inference for model (4.3).*

1. *Update $R_1, \ldots, R_{N-1}$: Sample from $\mathbb{P}(R_1, \ldots, R_{N-1} \mid J_N, {}^{(N)}\tau, \boldsymbol{K})$ using the HMC algorithm.*

2. *Update $J_N, {}^{(N)}\tau$: Sample from $\mathbb{P}(J_N, {}^{(N)}\tau \mid R_1, \ldots, R_{N-1}, \boldsymbol{K})$ using the Metropolis-Hastings algorithm.*

3. *Update $p_1, \ldots, p_N, e_N$: Apply the representations in (4.4).*

4.2 Bayesian nonparametric mixture model

Next, we develop a posterior inference scheme for model (4.1). To achieve an efficient MCMC sampling scheme, we recast the model completely in terms of random variables:

$$\begin{aligned} (X_j \mid \mathbf{Z}, \mathbf{K}) &\sim \mathcal{K}(X_j \mid Z_{K_j}), \;\; j = 1, \ldots, n, \qquad && \mathbf{Z} \sim \pi(\mathbf{Z}), \\ (K_j \mid \mathcal{P}_N) &\overset{iid}{\sim} \mathcal{P}_N = \sum_{i=1}^{N} p_i \delta_i(.) + e_N \delta_0(.), \qquad && \mathcal{P}_N \sim \text{N-NRMI}(\rho), \end{aligned} \tag{4.6}$$

where $\mathbf{Z} := (Z_0, Z_1, \ldots, Z_N)$ are the latent variables induced by the prior $\pi(\mathbf{Z})$, and $\mathbf{K} = (K_1, \ldots, K_n)$ are conditionally independent classifiers that relate $\mathbf{Z}$ to the latent variables Y_j, i.e., $Y_j = Z_{K_j}$, for $j = 1, \ldots, n$. The posterior of the model has the format $\mathbb{P}(p_1, \ldots, p_N, e_N, \mathbf{Z}, \mathbf{K} \mid \mathbf{X})$. To sample from it, we can devise a blocked Gibbs sampler that draws iteratively from the conditional distributions $\mathbb{P}(p_1, \ldots, p_N, e_N \mid \mathbf{K})$, $\mathbb{P}(\mathbf{Z} \mid \mathbf{K}, \mathbf{X})$ and $\mathbb{P}(\mathbf{K} \mid p_1, \ldots, p_N, e_N, \mathbf{Z}, \mathbf{X})$. Algorithm 4.2 can be used for the first distribution, while the sampling procedures for the other two distributions depend on the model specifications.

The posterior inference results of model (4.1) can be used to estimate the predictive density for a new observation. We denote by $f(X_{n+1} \mid \mathbf{X})$ the predictive density of X_{n+1} conditioning on the current observations $\mathbf{X} = \{X_1, \ldots, X_n\}$ and Y_{n+1} the latent variable of this new observation, then

$$\begin{aligned} f(X_{n+1} \mid \mathbf{X}) &= \int \mathcal{K}(X_{n+1} \mid Y_{n+1}) d\mathbb{P}(Y_{n+1} \mid \mathbf{X}) \\ &= \iint \mathcal{K}(X_{n+1} \mid Y_{n+1}) d\mathbb{P}(Y_{n+1} \mid \mathcal{P}_N) d\mathbb{P}(\mathcal{P}_N \mid \mathbf{X}). \end{aligned}$$

Based on the parametrisation of (4.6), the inner integral can be expressed as

$$\int \mathcal{K}(X_{n+1} \mid Y_{n+1}) d\mathbb{P}(Y_{n+1} \mid \mathcal{P}_N) = \sum_{i=1}^{N} p_i \mathcal{K}(X_{n+1} \mid Z_i) + e_N \mathcal{K}(X_{n+1} \mid Z_0). \tag{4.7}$$

Thus, the predictive density $f(X_{n+1} \mid \mathbf{X})$ can be estimated by averaging (4.7) over the posterior values from different iterations.

To illustrate the posterior inference scheme, we give an example using the normal-normal/inverse gamma model. The model is in the format of (2.1) with $\mathcal{K}(X_i \mid Y_i) \sim \mathcal{N}(X_i \mid \mu_i, \sigma_i)$, where μ_i and σ_i stand for the mean and variance of the normal mixture kernel, respectively. Their priors are given by $\mu \mid \sigma \sim \mathcal{N}(\theta_\mu, \sigma/\kappa_0)$ and $\sigma^{-1} \sim \text{Ga}(a_0, b_0)$, which explain the name of the model. Note that the normal distribution is parametrised

in the mean and variance, and the gamma distribution is parametrised in the shape and rate. Using (4.6), the model is approximated by

$$\begin{aligned}(X_j \mid \mu, \sigma, \mathbf{K}) &\sim \mathcal{N}(X_j \mid \mu_{K_j}, \sigma_{K_j}), \;\; j = 1, \ldots, n, \\ (\mu_i, \sigma_i^{-1} \mid \theta_\mu, \kappa_0, a_0, b_0) &\sim \mathcal{N}(\theta_\mu, \sigma_i/\kappa_0) \otimes \mathrm{Ga}(a_0, b_0), \;\; i = 0, 1, \ldots, N, \\ (K_j \mid \mathcal{P}_N) &\overset{iid}{\sim} \mathcal{P}_N = \sum_{i=1}^{N} p_i \delta_i(.) + e_N \delta_0(.), \\ \mathcal{P}_N &\sim \text{N-NRMI}(\rho).\end{aligned} \tag{4.8}$$

Next, we develop a blocked Gibbs sampler for the posterior of model (4.8). Note that the algorithm differs from the existing methods only in Step 1, and the derivation of Step 2, 3, and 4 can be found in, for example, Ishwaran and James (2002).

Algorithm 4.3. *Posterior inference for the normal-normal/inverse gamma model.*

1. *Update $p_1, \ldots, p_N, e_N$: Sample from $\mathbb{P}(p_1, \ldots, p_N, e_N \mid \boldsymbol{K})$ using Algorithm 4.2.*

2. *Update μ: Let $\{K_1^*, \ldots, K_m^*\}$ denote the current m unique values of $\boldsymbol{K}$. For each $j \in \{K_1^*, \ldots, K_m^*\}$, draw $\mu_j \leftarrow \mathcal{N}(\theta_j^*, \sigma_{Z_j}^*)$, where*

$$\theta_j^* = \sigma_{Z_j}^* \left(\theta_\mu/(\sigma_j/\kappa_0) + \sum_{\{i:K_i = K_j^*\}} X_i/\sigma_j \right), \;\; \sigma_{Z_j}^* = (n_j/\sigma_j + 1/(\sigma_j/\kappa_0))^{-1},$$

   and n_j is the number of times K_j^* occurs in $\boldsymbol{K}$. Also, for each $j \in \boldsymbol{K} - \{K_1^*, \ldots, K_m^*\}$, draw $\mu_j \leftarrow \mathcal{N}(\theta_\mu, \sigma_j/\kappa_0)$.

3. *Update σ: Let $\{K_1^*, \ldots, K_m^*\}$ denote the current m unique values of $\boldsymbol{K}$. For each $j \in \{K_1^*, \ldots, K_m^*\}$, draw*

$$\sigma_j^{-1} \sim Ga\left(a_0 + n_j/2, b_0 + \sum_{\{i:K_i = K_j^*\}} (X_i - \mu_j)^2/2 \right).$$

   Also, for each $j \in \boldsymbol{K} - \{K_1^*, \ldots, K_m^*\}$, draw $\sigma_j^{-1} \leftarrow Ga(a_0, b_0)$.

4. *Update $\boldsymbol{K}$: Draw K_i according to*

$$K_i \leftarrow \sum_{j=1}^{N} p_{j,i}^* \delta_j(.) + e_{N,i}^* \delta_0(.), \quad i = 1, \ldots, n,$$

   where

$$(p_{1,i}^*, \ldots, p_{N,i}^*, e_{N,i}^*) \propto \left(\frac{p_1}{\sqrt{\sigma_1}} e^{-\frac{(X_i - \mu_1)^2}{2\sigma_1}}, \ldots, \frac{p_N}{\sqrt{\sigma_N}} e^{-\frac{(X_i - \mu_N)^2}{2\sigma_N}}, \frac{e_N}{\sqrt{\sigma_0}} e^{-\frac{(X_i - \mu_0)^2}{2\sigma_0}} \right).$$

4.3 Caron-Fox model

Then, we discuss the finite approximation of the Caron-Fox model (2.2). We replace the CRM(ρ) prior by N-CRM(ρ) and obtain the truncated model Z_N in terms of

$$\begin{aligned} Z_N = \sum_{j=1}^{n} \delta_{\{s_j, r_j\}}, \qquad & s_j, r_j \overset{iid}{\sim} \frac{\Theta_N(.)}{\Theta_N(\mathcal{S})} = \sum_{i=1}^{N} \frac{J_i}{\tau}\delta_{K_i}(.) + \frac{^{(N)}\tau}{\tau}\delta_{K_0}(.), \\ n \sim \text{Pois}(\tau^2), \qquad & \Theta_N \sim \text{N-CRM}(\rho), \end{aligned} \tag{4.9}$$

where $K_1, \ldots, K_N$ are the N nodes with the highest sociabilities, and K_0 is the collection of the nodes with lower sociabilities. Denote by n_{ij} the number of edges from node K_i to node K_j, and set $m_i := \sum_{j=0}^{N}(n_{ij} + n_{ji}) > 0$ as the total number of edges sent or received by node K_i, for $i = 0, 1, \ldots, N$. Then, the posterior of (4.9) is given by

$$\begin{aligned} &\mathbb{P}(R_1 \in dr_1, \ldots, R_{N-1} \in dr_{N-1}, J_N \in dz, {}^{(N)}\tau \in dy \mid (m_i)_{0\le i\le N}) \\ &\propto z^{m_1+\cdots+m_N} r_1^{-m_1} \ldots r_{N-1}^{-(m_1+\cdots+m_{N-1})} y^{m_0} \exp(-(z\Pi^{-1}\Sigma + y)^2) e^{-\rho(z,\infty)}\rho(z) \\ &\quad \times \Pi^{-1}\left(\prod_{i=1}^{N-1}(z\Pi^{-1}r_1 \ldots r_{i-1})\rho(z\Pi^{-1}r_1 \ldots r_{i-1})\right) f_{\rho,z}(y)dydzdr_{N-1}\ldots dr_1. \end{aligned} \tag{4.10}$$

Note that the nodes $K_1, \ldots, K_N, K_0$ in the model (4.9) are not identical to those in the model (2.2). For example, the node K_1 in (4.9) has higher sociability than K_2 almost surely, but this is not true in (2.2). To approximate the posterior of (2.2) by (4.10), we need to assign the observed edges $\{\tilde{m}_i\}_{i=1}^{N_a}$ to the nodes $K_1, \ldots, K_N, K_0$ artificially. To this end, we set $m_1 > \cdots > m_N$ as the N largest terms of $\{\tilde{m}_i\}_{i=1}^{N_a}$, for $N \le N_a$, and m_0 as the sum of the smaller terms. It follows that a node with more edges has higher posterior sociability in our estimation. This might be inconsistent with the observation of (2.2). Therefore, the proposed approach is an approximation, rather than an exact representation, of the posterior of the original Caron-Fox model.

We remark that our characterisation is different from Li and Campbell (2021). The latter truncated the CRM prior at a fixed level and investigated the probability that all the active nodes were in the truncation region. Instead, we concentrate on the N most popular nodes, and there might be some active nodes outside the truncation region.

5 Examples

In this section, we demonstrate the results in the previous sections with the Lévy measure $\rho(dw) = tw^{-\alpha-1}e^{-\mu w}dw$, for $t > 0$. We will consider the following parametrisations: (i) $\alpha \in (0, 1)$, $\mu > 0$, (ii) $\alpha = 0$, $\mu > 0$, and (iii) $\alpha \in (0, 1)$, $\mu = 0$. We will discuss the simulation algorithms and posterior inference schemes in these cases.

5.1 Generalised gamma process

The generalised gamma process has become the most popular nonparametric prior in recent years. It can be derived from the Lévy measure $\rho(dw) = tw^{-\alpha-1}e^{-\mu w}dw$, for

$\alpha \in (0,1)$ and $\mu > 0$. The parametrisation of the Lévy measure is not unique. For example, Brix (1999) and Lijoi et al. (2007) formulated the Lévy measure as $\Gamma(1-\alpha)^{-1}w^{-\alpha-1}e^{-\mu w}dw$. This can be achieved by setting $t = \Gamma(1-\alpha)^{-1}$. It is also worth mentioning the remarkable Proposition 2 of Lijoi et al. (2007), which showed that the distribution of the number of distinct clusters induced by a NGG process prior relies on α and $\beta := \mu^\alpha/\alpha$ only. A bigger β leads to more distinct clusters, and a large value of α yields a flatter distribution on the number of distinct clusters.

To obtain the results of Lemma 3.1, we need to derive the density of the generalised gamma process. To this end, we use the Zolotarev integral to invert its Laplace transform. It follows that

$$\begin{aligned} f_\rho(z) =& \mathcal{L}^{-1}\{\mathbb{E}(e^{-\beta\tau})\} = \mathcal{L}^{-1}\{\exp(-t\Gamma(1-\alpha)\alpha^{-1}\left((\beta+\mu)^\alpha - \mu^\alpha\right))\} \\ =& \exp(\mu^\alpha T - \mu z)\frac{1}{\pi}\int_0^\pi \frac{\alpha}{1-\alpha}A(u)z^{-\frac{1}{1-\alpha}}T^{\frac{1}{1-\alpha}}\exp(-A(u)z^{-\frac{\alpha}{1-\alpha}}T^{\frac{1}{1-\alpha}})du, \end{aligned}$$

where $T := t\Gamma(1-\alpha)\alpha^{-1}$ and

$$A(u) := \left(\sin(\alpha u)^\alpha \sin((1-\alpha)u)^{1-\alpha}\sin(u)^{-1}\right)^{1/(1-\alpha)}. \tag{5.1}$$

Then, the results of Lemma 3.1 and Theorem 3.2 follow immediately.

To sample from the N largest jumps of the generalise gamma process, all the algorithms introduced in Section 3 could be used. For the rejection algorithm, we choose the envelope Lévy measure $\rho_0(dw) := tw^{-\alpha-1}dw$. Since $\int_x^\infty \rho_0(dw) = (t/\alpha)x^{-\alpha}$, the tail distribution of ρ_0 can be inverted in close form, and the i-th largest jump J_i is accepted with the probability $\exp(-\mu J_i)$. See Godsill and Kındap (2022) for a further discussion. To simulate the N-trimmed generalised gamma process, we rewrite its conditional Lévy-Khintchine representation as

$$\mathbb{E}\left(\exp(-\beta\,{}^{(N)}\tau) \mid J_1,\ldots,J_N\right) = \exp\left(-tJ_N^{-\alpha}\int_0^1 (1-e^{-\beta J_N w})w^{-\alpha-1}e^{-\mu J_N w}dw\right).$$

It follows that ${}^{(N)}\tau \overset{d}{=} J_N \times Z_{\alpha,\mu J_N, tJ_N^{-\alpha}}$, where $Z_{\alpha,\mu,t}$ denotes a truncated tempered stable process with Lévy measure $tw^{-\alpha-1}e^{-\mu w}\mathbb{1}_{\{0<w<1\}}dw$, and the exact simulation algorithm for such a process can be found in Dassios et al. (2020). We attach the full steps in the supplementary material (Zhang and Dassios, 2024).

Next, we consider the finite approximation to the NGG process hierarchical model. From (4.5) we know the model posterior is characterised by

$$\begin{aligned} &\mathbb{P}(R_1 \in dr_1,\ldots,R_{N-1}\in dr_{N-1}, J_N \in dz, {}^{(N)}\tau \in dy \mid \mathbf{K}) \\ &\propto z^{(n_1+\cdots+n_N)-N\alpha-1}r_1^{-n_1+\alpha-1}\ldots r_{N-1}^{-(n_1+\cdots+n_{N-1})+(N-1)\alpha-1}y^{n_0}(z\Pi^{-1}\Sigma+y)^{-n} \\ &\quad\times\exp\left(-t\int_z^\infty w^{-\alpha-1}e^{-\mu w}dw\right)\exp(-z\mu\Pi^{-1}\Sigma)f_{\rho,z}(y)dydzdr_1\ldots dr_{N-1}. \end{aligned}$$

To sample from this distribution, we first use the HMC algorithm to draw from

$$\begin{aligned}&\mathbb{P}(R_1 \in dr_1, \ldots, R_{N-1} \in dr_{N-1} \mid J_N = z, {}^{(N)}\tau = y, \mathbf{K})\\ &\propto (z\Pi^{-1}\Sigma + y)^{-n} \exp(-z\mu\Pi^{-1}\Sigma) \prod_{i=1}^{N-1} r_i^{-(n_1+\cdots+n_i)+i\alpha-1}.\end{aligned}$$

The gradient of the log-posterior, after the change of variable $\mathcal{R}_i := \tan(\pi(r_i - 0.5))$, is given by

$$\begin{aligned}&\frac{d}{d\mathcal{R}_i}\log(\mathbb{P}(\mathcal{R}_1, \ldots, \mathcal{R}_{N-1} \mid J_N = z, {}^{(N)}\tau = y, \mathbf{K}))\\ &= -\frac{nz}{z\Pi^{-1}\Sigma + y}\left(\Pi^{-1}\frac{d}{d\mathcal{R}_i}\Sigma - \Sigma\Pi^{-2}\frac{d}{d\mathcal{R}_i}\Pi\right) - z\mu\left(\Pi^{-1}\frac{d}{d\mathcal{R}_i}\Sigma - \Sigma\Pi^{-2}\frac{d}{d\mathcal{R}_i}\Pi\right)\\ &\quad + \frac{-(n_1+\cdots+n_i)+i\alpha-1}{0.5 + (1/\pi)\arctan(\mathcal{R}_i)}\frac{1}{\pi}\frac{1}{1+\mathcal{R}_i^2} - \frac{2\mathcal{R}_i}{1+\mathcal{R}_i^2}\end{aligned}$$

where

$$\frac{d}{d\mathcal{R}_i}\Pi = \Pi\frac{1}{\pi}\frac{1}{1+\mathcal{R}_i^2}\left(\frac{1}{2} + \frac{1}{\pi}\arctan(\mathcal{R}_i)\right)^{-1},$$

and

$$\frac{d}{d\mathcal{R}_i}\Sigma = \sum_{j=i}^{N-1}\left[\prod_{k=1}^{i-1}\left(\frac{1}{2} + \frac{1}{\pi}\arctan(\mathcal{R}_k)\right)\right]\frac{1}{\pi}\frac{1}{1+\mathcal{R}_i^2}\left[\prod_{k=i+1}^{j}\left(\frac{1}{2} + \frac{1}{\pi}\arctan(\mathcal{R}_k)\right)\right].$$

Then, we use the Metropolis-Hastings algorithm to sample from

$$\begin{aligned}&\mathbb{P}(J_N \in dz, {}^{(N)}\tau \in dy \mid R_1 = r_1, \ldots R_{N-1} = r_{N-1}, \mathbf{K})\\ &\propto \frac{z^{(n_1+\cdots+n_N)-N\alpha-1}y^{n_0}}{(z\Pi^{-1}\Sigma + y)^n}\exp\left(-t\int_z^\infty w^{-\alpha-1}e^{-\mu w}dw\right)\exp(-z\mu\Pi^{-1}\Sigma)f_{\rho,z}(y),\end{aligned}$$

with the transition kernels $\tilde{z} \leftarrow \text{log-normal}(\tilde{z}; \log(z), \sigma_0)$ and $\tilde{y} \leftarrow f_{\rho,\tilde{z}}(\tilde{y})$. By running these algorithms iteratively, we get the posterior values of $(R_1, \ldots, R_{N-1}, J_N, {}^{(N)}\tau)$. Then, we can derive the posterior probability masses according to (4.4).

Finally, the posterior distribution of the network model (4.9) is obtained by inputting the Lévy measure of the generalised gamma process into (4.10). The posterior expression and blocked Gibbs sampler can be found in the supplementary material.

5.2 Gamma process

The gamma process is the most fundamental Bayesian nonparametric prior. Its Lévy measure $\rho(dw) = tw^{-1}e^{-\mu w}dw$ is a special case of that of the generalised gamma process with $\alpha = 0$. Without loss of generality, we set $\mu = 1$. To validate Lemma 3.1 and Theorem 3.2, we derive the density of a gamma process τ as

$$f_\rho(z) = \mathcal{L}^{-1}\{\mathbb{E}(e^{-\beta\tau})\} = \mathcal{L}^{-1}\{(1+\beta)^{-t}\} = z^{t-1}e^{-z}/\Gamma(t).$$

To simulate the N largest jumps of the gamma process, we choose the rejection algorithm with the envelope Lévy measure $\rho_0(dw) = tw^{-1}(1+w)^{-1}dw$, whose tail distribution is $t\ln(w^{-1}+1)$. The acceptance rate is very high in this case. In fact, Rosiński (2001) (see also Campbell et al. 2019 and Godsill and Kındap 2022) showed that the average number of rejected samples is $\alpha\nu$, where ν is the Euler Mascheroni constant. Therefore, to sample from the N largest jumps, we only need to generate $(N+\alpha\nu)$ jumps from the envelope in average. To simulate the N-trimmed gamma process, we rewrite its conditional Lévy-Khintchine representation as

$$\mathbb{E}\left(\exp(-\beta\,{}^{(N)}\tau) \mid J_1,\ldots,J_N\right) = \exp\left(-t\int_0^1 (1-e^{-\beta J_N w})w^{-1}e^{-J_N w}dw\right).$$

It follows that ${}^{(N)}\tau \overset{d}{=} J_N \times Z_{J_N,t}$, where $Z_{\mu,t}$ denotes a truncated gamma process with Lévy measure $tw^{-1}e^{-w}\mathbb{1}_{\{0<w<1\}}dw$, and the exact simulation algorithm for such a process can be found in Dassios et al. (2019).

The posterior distribution and inference scheme are same as before with $\alpha = 0$ and $\mu = 1$; we omit the details. We remark that the blocked Gibbs sampler in the current paper is different from the one in Zhang and Dassios (2023). In the current work, we use the HMC algorithm to update $R_1,\ldots,R_{N-1}$ only, instead of $J_1, R_1,\ldots,R_{N-1}$ as in Zhang and Dassios (2023). This modification removes the density of the N-trimmed gamma process from the HMC algorithm, thus significantly improves the performance of the blocked Gibbs sampler.

5.3 Stable process

Another important class of CRMs is the stable process induced by the Lévy measure $\rho(dw) = tw^{-\alpha-1}dw$. We first derive the density of the stable process via the Zolotarev integral,

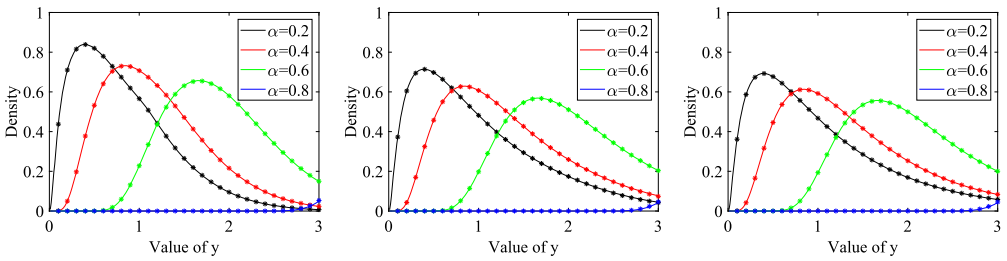
$$\begin{aligned} f_\rho(z) =& \mathcal{L}^{-1}\{\mathbb{E}(e^{-\beta\tau})\} = \mathcal{L}^{-1}\{\exp(-t\beta^\alpha\Gamma(1-\alpha))\} \\ =& \frac{1}{\pi}\int_0^\pi \frac{\alpha}{1-\alpha}A(u)z^{-\frac{1}{1-\alpha}}T^{\frac{1}{1-\alpha}}\exp(-A(u)z^{-\frac{\alpha}{1-\alpha}}T^{\frac{1}{1-\alpha}})du, \end{aligned}$$

where $T := t\Gamma(1-\alpha)$, and $A(u)$ is defined as (5.1).

Since the tail distribution of $\rho(dw)$ has an explicit expression $t\int_x^\infty w^{-\alpha-1}dw = (t/\alpha)x^{-\alpha}$, the ILM algorithm is preferable. To simulate the N-trimmed stable process, we rewrite its conditional Lévy-Khintchine representation as

$$\mathbb{E}\left(\exp(-\beta\,{}^{(N)}\tau) \mid J_1,\ldots,J_N\right) = \exp\left(-tJ_N^{-\alpha}\int_0^1 (1-e^{-\beta J_N w})w^{-\alpha-1}dw\right).$$

It follows that ${}^{(N)}\tau \overset{d}{=} J_N \times Z_{\alpha,tJ_N^{-\alpha}}$, where $Z_{\alpha,t}$ denotes a truncated stable process with Lévy measure $tw^{-\alpha-1}\mathbb{1}_{\{0<w<1\}}dw$, and the exact simulation algorithm for such a process can be found in Dassios et al. (2020). The posterior distribution and inference scheme are same as the generalised gamma process with $\mu = 0$; we omit the details.



(a) $z = 1, \mu = 1, t = 1$. (b) $z = 2, \mu = 1, t = 1$. (c) $z = 3, \mu = 1, t = 1$.

Figure 1: Density of the N-trimmed generalised gamma process and the numerical inverse Laplace transform results.

6 Numerical implementation

In this section, we present some numerical results for the simulation algorithms and posterior inference schemes. Our implementation will focus on the generalised gamma process and its normalisation. The analysis was carried out on Matlab 2023a on a 64-bit Windows desktop with an Intel i9-12900 processor and 64GB RAM.

6.1 Density of the N-trimmed generalised gamma process

To illustrate the results of Lemma 3.1, we plot the density $f_{\rho,z}(y)$ with different parameters in Figure 1. We also apply the concentrated matrix-exponential functions method (CME, see Horváth et al. 2020) to invert the Laplace transform of ${}^{(N)}\tau$ numerically at some fixed points. The results are denoted by stars. From Figure 1 we can see that both methods produce similar results.

6.2 Simulation algorithms

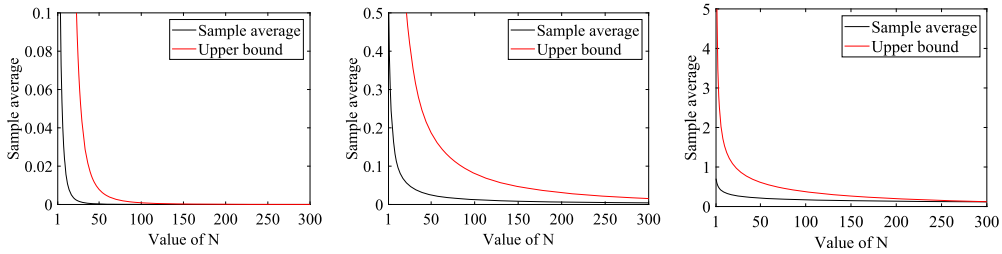
We use the ILM, rejection, AR and MCMC algorithms introduced in Section 3 to sample from the five largest jumps and the sum of smaller jumps of the generalised gamma process. The sample averages of the jumps are recorded in Table 1. From the table, we can see that all of these algorithms can achieve a reasonable Monte Carlo accuracy as long as the sample size is large enough. However, when α is large, the rejection algorithm becomes slow due to a low acceptance rate.

6.3 Truncation error

To illustrate the truncation error of the generalised gamma process, we sample from ${}^{(N)}\tau$ with different truncation levels and present the sample averages in Figure 2. We also compare the sample averages to the upper bound of the truncation error given in Appendix A of Brix (1999). The figures show that for the same μ and t, the truncation error converges to zero slower when α is large. In addition, we sample from the truncation

	Algorithm	J_1	J_2	J_3	J_4	J_5	${}^{(N)}\tau$	sum	Time
	true mean	0.6188	0.2228	0.1028	0.0531	0.0288	0.0413	1.0676	N.A.
$\alpha = 0.1$	ILM	0.6142	0.2226	0.1026	0.0524	0.0284	0.0418	1.0619	102
$\mu = 1.0$	Rejection	0.6253	0.2248	0.1042	0.0533	0.0288	0.0420	1.0784	8
$t = 1.0$	AR	0.6197	0.2193	0.1033	0.0530	0.0290	0.0409	1.0652	103
	MCMC	0.6238	0.2209	0.1029	0.0527	0.0282	0.0407	1.0691	105
	true mean	0.6150	0.2490	0.1334	0.0820	0.0536	0.1659	1.2989	N.A.
$\alpha = 0.3$	ILM	0.6065	0.2476	0.1310	0.0805	0.0526	0.1639	1.2821	79
$\mu = 1.0$	Rejection	0.6156	0.2467	0.1348	0.0816	0.0530	0.1647	1.2963	5
$t = 1.0$	AR	0.6150	0.2484	0.1327	0.0812	0.0531	0.1655	1.2959	83
	MCMC	0.6188	0.2502	0.1317	0.0795	0.0520	0.1647	1.2969	90
	true mean	0.6128	0.2730	0.1642	0.1123	0.0820	0.5311	1.7754	N.A.
$\alpha = 0.5$	ILM	0.6025	0.2725	0.1636	0.1115	0.0809	0.5273	1.7583	85
$\mu = 1.0$	Rejection	0.6114	0.2768	0.1653	0.1127	0.0813	0.5291	1.7766	6
$t = 1.0$	AR	0.6053	0.2712	0.1635	0.1107	0.0806	0.5273	1.7585	88
	MCMC	0.6068	0.2699	0.1621	0.1111	0.0806	0.5254	1.7558	107
	true mean	0.6135	0.3000	0.1936	0.1420	0.1068	1.6579	3.0138	N.A.
$\alpha = 0.7$	ILM	0.6032	0.2990	0.1938	0.1411	0.1099	1.6366	2.9837	105
$\mu = 1.0$	Rejection	0.6114	0.2966	0.1933	0.1408	0.1093	1.6325	2.9840	19
$t = 1.0$	AR	0.6094	0.2949	0.1929	0.1410	0.1096	1.6364	2.9842	109
	MCMC	0.6094	0.2978	0.1945	0.1417	0.1093	1.6292	2.9819	142
	true mean	0.6134	0.3226	0.2212	0.1711	0.1349	7.9117	9.3749	N.A.
$\alpha = 0.9$	ILM	0.6208	0.3266	0.2239	0.1708	0.1380	8.0475	9.5275	176
$\mu = 1.0$	Rejection	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	$> 10^4$
$t = 1.0$	AR	0.6146	0.3231	0.2220	0.1707	0.1383	8.0479	9.5166	179
	MCMC	0.6192	0.3254	0.2234	0.1706	0.1383	8.0409	9.5178	233

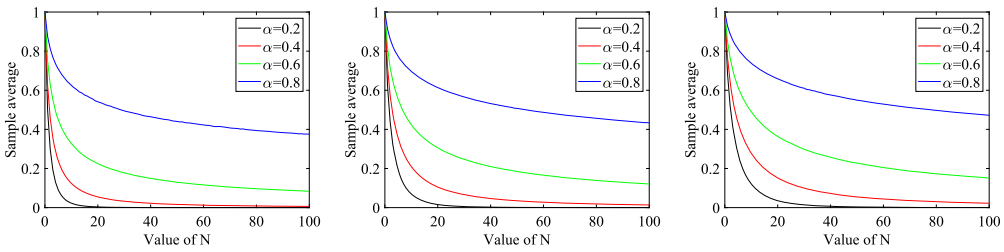
Table 1: Sample averages of the five largest jumps and the sum of smaller jumps of the generalised gamma process, the sample size is 10^4. The running times are in seconds and rounded to the nearest integer. Parallel computing was not used in the simulation.



(a) $\alpha = 0.25$. (b) $\alpha = 0.5$. (c) $\alpha = 0.75$.

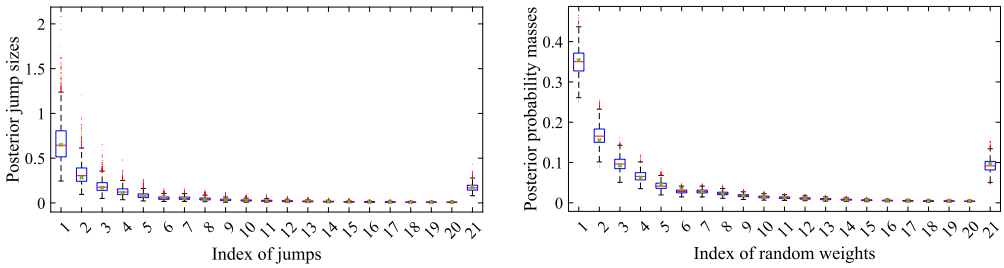
Figure 2: Sample average of the truncation error ${}^{(N)}\tau$ of the generalised gamma process and the upper bound, $\mu = 1$, $t = 1/\Gamma(1-\alpha)$.

error e_N of the NGG process and present the sample averages in Figure 3. From the figures we can see that the truncation error becomes larger as α and t increase. Also, a higher truncation level N leads to a smaller truncation error.



(a) $\mu = 1, t = 1$. (b) $\mu = 1, t = 2$. (c) $\mu = 1, t = 3$.

Figure 3: Sample average of the truncation error e_N of the NGG process.



(a) Posterior jump sizes. (b) Posterior probability masses.

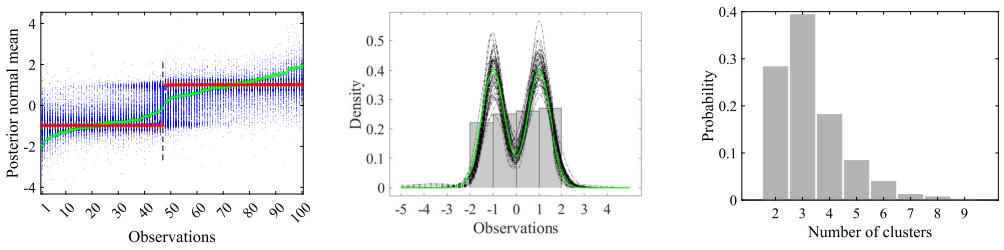
Figure 4: Posterior inference results of the hierarchical model (4.3) based on $n = 200$ observations. The true values of the variables are denoted by green cross. The parameters are set to $\alpha = 0.5$, $\mu = 1$, $t = 1$ and $N = 20$.

6.4 Blocked Gibbs sampler

To illustrate the performance of the blocked Gibbs sampler, we simulate a sequence of observations from the hierarchical model (4.3) and use Algorithm 4.2 to sample from the posterior of the model. The numerical results are presented in Figure 4. From the figures we can see that the blocked Gibbs sampler estimates the jump sizes $(J_1, \dots J_N, {}^{(N)}\tau)$ and the probability masses $(p_1, \dots, p_N, e_N)$ accurately.

6.5 Posterior inference of mixture model

Consider a bimodal mixture with the underlying density $f(x_i) = (1/2)\mathcal{N}(x_i; -1, 0.5^2) + (1/2)\mathcal{N}(x_i; 1, 0.5^2)$. We simulate $n = 100$ observations from the bimodal mixture, then use Algorithm 4.3 to estimate the underlying distribution. We follow the parametrisation in Section 3 of Lijoi et al. (2007) and use the pair of parameters (α, β). In the HMC step, we use the leapfrog steps $L = 10$ and adjust the step size ϵ to obtain an acceptance rate of around 0.6. The posterior mean values are presented in Figure 5(a). From the figure we can see that the posterior induces two modes for the model at -1 and 1. Then we randomly select fifty iterations and use their posterior values to plot the predictive



(a) Posterior mean values. (b) Predictive densities. (c) Occupied clusters.

Figure 5: Posterior inference results of the bimodal mixture with $n = 100$ observations. Each observation X_i is sampled from a normal distribution with mean $\{-1, 1\}$ with equal probabilities and variance 0.5^2. The plot is based on 5000 sampled values following an initial 5000 iteration burn-in using Algorithm 4.3. The parameters are set to $\alpha = 0.25$, $\beta = 0.5$ and $N = 50$. (a) Posterior mean values. The values of X_i and μ_i are denoted by green cross and red plus, respectively. (b) Fifty randomly selected predictive densities. The observations are presented by the histogram, and the true mixture density is denoted by the green curve. (c) Proportion of occupied clusters.

density (4.7) in Figure 5(b). We also count the numbers of occupied clusters and record their proportion in Figure 5(c). In addition, we use Algorithm 4.3 to analyse the galaxy velocity data and present the mean predictive density in the supplementary material.

Next, we compare the performance of our posterior inference scheme with the celebrated BNPdensity package developed by Arbel et al. (2021). The package facilitates the posterior inference method in James et al. (2009) and Barrios et al. (2013). We draw $n = 100$ independent samples from the bimodal mixture, then analyse the data using Algorithm 4.3 and the BNPdensity package. Their performance are monitored by the convergence of the number of occupied clusters and the deviance of the estimated density. For the posterior inference results from the r-th iteration, we denote by $K^{(r)}$ the number of occupied clusters and $n_j^{(r)}$ the size of each occupied cluster, such that $\sum_{j=1}^{K^{(r)}} n_j^{(r)} = n$. The deviance is a function of all estimated parameters defined as

$$D^{(r)} := -2 \sum_{i=1}^{n} \log \left(\sum_{j=1}^{K^{(r)}} \frac{n_j^{(r)}}{n} \mathcal{K}(X_i \mid Y_j^{(r)}) \right).$$

These quantities have been used in the comparison study of the existing literature. See, for example, Neal (2000), Papaspiliopoulos and Roberts (2008), Kalli et al. (2011) and Canale et al. (2022). The efficiency of the posterior inference scheme can be evaluated by calculating the integrated autocorrelation time (IAT) and effective sample size (ESS) of these quantities. The IAT is defined as $\tau := 0.5 + \sum_{l=1}^{\infty} \rho_l$ (see Sokal 1997), where ρ_l is the autocorrelation at lag l. It illustrates the statistical error of the target function in Monte Carlo estimation. The difficulty of calculating τ arises from the covariance between the states, which have been used to evaluate ρ_l. Sokal (1997) suggested using

	$\hat{\tau}_K$	$\hat{\tau}_D$	$\bar{K}$	$\bar{D}$	E_K	E_D	Size
Algorithm 4.3 ($N = 50$)	14.0633	1.1112	4.1860	278.7610	504	4348	20134
Algorithm 4.3 ($N = 100$)	11.4309	0.8962	4.0110	278.0777	263	1561	6894
BNPdensity ($\epsilon = 0.01$)	9.9922	0.7194	6.3845	280.7044	449	8039	9699
BNPdensity ($\epsilon = 0.005$)	9.2284	0.5366	7.0553	280.7474	441	7349	9521

Table 2: Posterior inference results of the bimodal mixture with sample size $n = 100$, $\alpha = 0.25$, $\beta = 0.5$, $t = 1/\Gamma(1-\alpha)$, the running time is 200 seconds. K and D stand for the number of occupied clusters and deviance, respectively. $\hat{\tau}$ represents the estimated IAT, and E stands for the ESS. The ESS is rounded to the nearest integer.

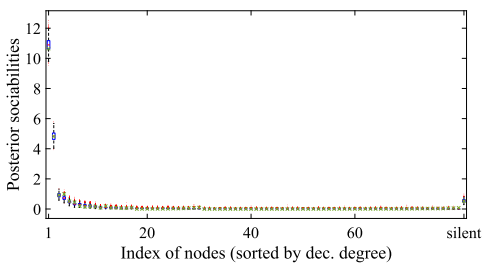
	$\hat{\tau}_K$	$\hat{\tau}_D$	K	D	E_K	E_D	Size
Algorithm 4.3 ($N = 50$)	12.2177	1.3663	8.3615	280.2764	209	2795	18750
Algorithm 4.3 ($N = 100$)	7.6820	1.4852	8.4865	279.6918	202	590	6344
BNPdensity ($\epsilon = 0.01$)	7.0525	0.6113	10.7765	280.2148	495	4433	8064
BNPdensity ($\epsilon = 0.005$)	5.9858	0.7776	10.8190	280.0456	763	3209	8023

Table 3: Posterior inference results of the bimodal mixture with sample size $n = 100$, $\alpha = 0.75$, $\beta = 0.5$, $t = 1/\Gamma(1-\alpha)$, the running time is 200 seconds.

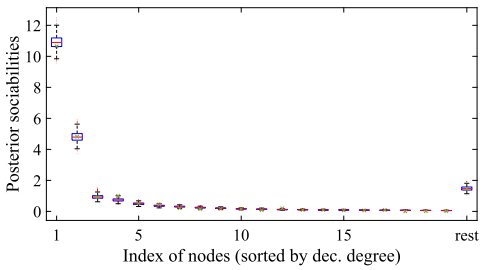
the estimator $\hat{\tau} = 0.5 + \sum_{l=1}^{C-1} \hat{\rho}_l$ for τ, where $\hat{\rho}_l$ is the estimated autocorrelation at lag l, and C is the user-specified cut-off point. We use the same cut-off point as Kalli et al. (2011), i.e., $C := \min\{l : |\hat{\rho}_l| < 2/\sqrt{M}\}$, where M is the number of iterations. This makes the cut-off point C the smallest lag for which we would not reject the null hypothesis $H_0 : \rho_l = 0$. On the other hand, the ESS measures the number of effective samples in the posterior inference results. Due to the autocorrelation, the ESS would be smaller than the length of the Markov chain, and a higher ESS implies a better estimation. In practice, the ESS can be computed by the CODA package. See Plummer et al. (2006) for more details. The numerical results are recorded in Table 2 and Table 3. To achieve a fair comparison, we run both methods for 200 seconds and record the number of iterations. From the tables we can see that Algorithm 4.3 leads to fewer occupied clusters but a higher IAT. Also, the efficiency of Algorithm 4.3 is very sensitive to the truncation level. This is mainly caused by the exact simulation of the truncated generalised gamma process. We refer to Dassios et al. (2020) for a further discussion about its performance. In addition, we analyse the galaxy velocity data with both methods and attach the results in the supplementary material.

6.6 Posterior inference of Caron-Fox model

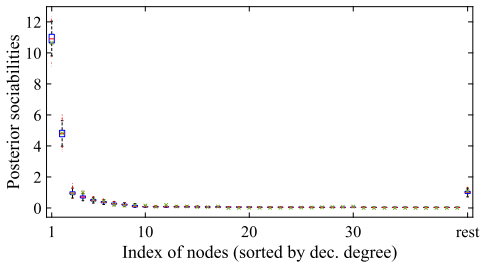
To demonstrate the approximation of the Caron-Fox model, we first sample from the model (2.2) using the algorithm in Section 5.5 of Caron and Fox (2017). The simulation is based on the generalised gamma process prior with hyperparameters $\alpha = 0.5, \mu = 0.1, t = 5$. The simulated total sociability is $\tau = 21.1928$, leading to 80 active nodes and 466 edges. Based on the observations, we estimate the sociabilities of the nodes using the original posterior sampler of Caron and Fox (2017) and our approximation (4.10). We present the numerical results in Figure 6. From the figures, we can see that both methods can recover the sociabilities of the nodes accurately. However, we find that



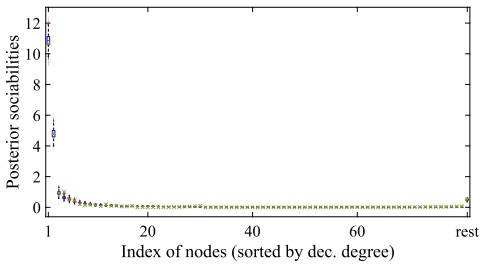
(a) Caron-Fox sampler, running time: 5 secs.



(b) $N = 20$, running time: 104 secs.



(c) $N = 40$, running time: 158 secs.



(d) $N = 80$, running time: 344 secs.

Figure 6: Posterior inference results of the Caron-Fox model and the running time. The true sociabilities are denoted by green cross. The plot is based on 2500 sampled values following an initial 2500 iteration burn-in. (a) The original posterior sampler in Caron and Fox (2017). (b)(c)(d) Posterior inference results based on the approximation (4.10).

the original posterior sampler is more efficient than ours. This is mainly caused by the exact simulation algorithm (Dassios et al., 2020) of the truncated generalised gamma process, whose efficiency is sensitive to the truncation level and could be low. On the other hand, Caron and Fox (2017) estimated the total sociability of the silent nodes by sampling from the exponentially tilted stable distribution, and the simulation algorithm (Devroye, 2009; Hofert, 2011) is extremely efficient. The details of the posterior inference scheme and additional numerical results can be found in the supplementary material.

7 Discussion

In this paper, we introduce a finite approximation to the CRM by keeping its N largest atom weights unchanged and combining the smaller atom weights into a single term. We call this finite-dimensional approximation the truncated Ferguson-Klass representation. We develop the simulation algorithms for the approximation and devise a blocked Gibbs sampler for posterior inference. Then, we adapt the approximation into the Bayesian nonparametric mixture model and design the posterior inference scheme. We also study the application of the approximation in the Caron-Fox model. Examples and numerical implementations are given based on the generalised gamma, gamma and stable processes. We remark that the approximation can also be applied to the celebrated beta

process. In the supplementary material, we provide an example based on the truncated Dickman process, which is a special case of the beta process.

The current work uses a user-specified truncation level N. It is possible to develop an adaptive truncation method (Griffin 2016) for our approximation, which allows the truncation level to be decided by the algorithm itself. In this case, we can assess the convergence of the samples by calculating the ESS and run the algorithm until the absolute value of the difference between two consecutive ESS values is lower than a threshold. Alternatively, we can develop a moment-matching version (Arbel and Prünster 2017) of the approximation. This method assesses the approximation quality by evaluating the discrepancy between the actual moments of the CRM and the empirical moments. The truncation level is then selected such that the discrepancy does not exceed a given threshold.

As explained in Section 4, the proposed approach is an approximation, rather than an exact representation, of the original Caron-Fox model. The gap between the original model and our approximation is the artificial assignment of the observed edges. For the directed Caron-Fox model, this could be improved by introducing a discrete latent variable, which indicates the assignment of $\{\tilde{m}_i\}_{i=1}^{N_a}$ to $K_1, \ldots, K_N, K_0$. Given the sociabilities $J_1, \ldots, J_N, {}^{(N)}\tau$, the distribution of the latent variable could be specified by a collection of multinomial probabilities. Then, we can update the sociabilities and the latent variable iteratively. For the undirected model, we follow the strategy in Caron and Fox (2017) and update the latent edges in each iteration of the inference algorithm.

From the numerical experiments in Section 6, we can see that apart from having the flexibility in the number of nodes to be estimated, the proposed approach does not add too much value to the original posterior sampler in Caron and Fox (2017). However, our method might be useful to the extensions of the Caron-Fox model. For example, Todeschini et al. (2020) considered an extension of the Caron-Fox model with overlapping community structures. The simulation of the model involves sampling from the jumps of the generalised gamma process, and truncation is required. The authors considered the ϵ-approximation of the Lévy measure in terms of $\rho^{\epsilon}(dw) := \rho(dw)\mathbb{1}_{\{w>\epsilon\}}$, such that the approximation process has finite activity almost surely. It would be interesting to use the N-CRM(ρ) approximation instead and compare the performance of the two methods. Similar applications may also be found in other extensions of the Caron-Fox model, such as Herlau et al. (2016) and Ricci et al. (2022). In the current work, the Caron-Fox model is used as a numerical illustration, and the emphasis has been given to the posterior inference scheme of the N-CRM(ρ) prior. For the purpose of network analysis, we will address the issues discussed above in future work.

Acknowledgments

We are grateful to the editor and two reviewers for giving us detailed and constructive comments. The comments have helped us a lot in improving the manuscript. We also wish to thank Prof. Igor Prünster for kindly reading the draft and giving us helpful suggestions.

Supplementary Material

Supplementary Material of "Posterior sampling from truncated Ferguson-Klass representation of normalised completely random measure mixtures" (DOI: 10.1214/24-BA1421SUPP; .pdf). (i) Derivation of Lemma 3.1 and Theorem 3.2, (ii) example of the truncated Dickman process, (iii) exact simulation algorithm for truncated tempered stable process, (iv) additional numerical experiments of the mixture model, (v) posterior inference scheme for the Caron-Fox model and additional numerical results.

References

Arbel, J., De Blasi, P., and Prünster, I. (2019). "Stochastic approximations to the Pitman-Yor process." *Bayesian Analysis*, 14(4): 1201–1219. MR4136558. doi: https://doi.org/10.1214/18-BA1127. 2

Arbel, J., Kon Kam King, G., Lijoi, A., Nieto-Barajas, L., and Prünster, I. (2021). "BNPdensity: Bayesian nonparametric mixture modelling in R." *Australian & New Zealand Journal of Statistics*, 63(3): 542–564. MR4374512. 3, 22

Arbel, J. and Prünster, I. (2017). "A moment-matching Ferguson & Klass algorithm." *Statistics and Computing*, 27(1): 3–17. MR3598905. doi: https://doi.org/10.1007/s11222-016-9676-8. 2, 25

Argiento, R., Bianchini, I., and Guglielmi, A. (2016a). "A blocked Gibbs sampler for NGG-mixture models via a priori truncation." *Statistics and Computing*, 26(3): 641–661. MR3489862. doi: https://doi.org/10.1007/s11222-015-9549-6. 2

Argiento, R., Bianchini, I., and Guglielmi, A. (2016b). "Posterior sampling from ε-approximation of normalized completely random measure mixtures." *Electronic Journal of Statistics*, 10(2): 3516–3547. 2

Barrios, E., Lijoi, A., Nieto-Barajas, L. E., and Prünster, I. (2013). "Modeling with normalized random measure mixture models." *Statistical Science*, 28(3): 313–334. MR3135535. doi: https://doi.org/10.1214/13-STS416. 1, 22

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. MR2325310. 2

Brix, A. (1999). "Generalized gamma measures and shot-noise Cox processes." *Advances in Applied Probability*, 31(4): 929–953. MR1747450. doi: https://doi.org/10.1239/aap/1029955251. 1, 16, 19

Broderick, T., Jordan, M. I., and Pitman, J. (2012). "Beta processes, stick-breaking and power laws." *Bayesian Analysis*, 7(2): 439–475. MR2934958. doi: https://doi.org/10.1214/12-BA715. 1

Campbell, T. (2016). *Truncated Bayesian nonparametrics*. ProQuest LLC, Ann Arbor, MI. Thesis (Ph.D.)–Massachusetts Institute of Technology. MR3641161. 10

Campbell, T., Huggins, J. H., How, J. P., and Broderick, T. (2019). "Truncated random measures." *Bernoulli*, 25(2): 1256–1288. MR3920372. doi: https://doi.org/10.3150/18-bej1020. 6, 9, 10, 18

Canale, A., Corradin, R., and Nipoti, B. (2022). "Importance conditional sampling for Pitman-Yor mixtures." *Statistics and Computing*, 32(3): Paper No. 40, 18. MR4423541. doi: https://doi.org/10.1007/s11222-022-10096-0. 22

Caron, F. and Fox, E. B. (2017). "Sparse graphs using exchangeable random measures." *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 79(5): 1295–1366. MR3731666. doi: https://doi.org/10.1111/rssb.12233. 3, 23, 24, 25

Dassios, A., Lim, J. W., and Qu, Y. (2020). "Exact simulation of a truncated Lévy subordinator." *ACM Transactions on Modeling and Computer Simulation*, 30(3). MR4122822. doi: https://doi.org/10.1145/3368088. 16, 18, 23, 24

Dassios, A., Qu, Y., and Lim, J. W. (2019). "Exact simulation of generalised Vervaat perpetuities." *Journal of Applied Probability*, 56(1): 57–75. MR3981146. doi: https://doi.org/10.1017/jpr.2019.6. 18

De Blasi, P., Favaro, S., Lijoi, A., Mena, R. H., Prunster, I., and Ruggiero, M. (2015). "Are Gibbs-type priors the most natural generalization of the Dirichlet process?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2): 212–229. 2

Devroye, L. (2009). "Random variate generation for exponentially and polynomially tilted stable distributions." *ACM Transactions on Modeling and Computer Simulation*, 19(4). 24

Favaro, S. and Teh, Y. W. (2013). "MCMC for normalized random measure mixture models." *Statistical Science*, 28(3): 335–359. MR3135536. doi: https://doi.org/10.1214/13-STS422. 2

Ferguson, T. S. (1973). "A Bayesian analysis of some nonparametric problems." *The Annals of Statistics*, 1: 209–230. MR0350949. 1

Ferguson, T. S. and Klass, M. J. (1972). "A representation of independent increment processes without Gaussian components." *Annals of Mathematical Statistics*, 43: 1634–1643. MR0373022. doi: https://doi.org/10.1214/aoms/1177692395. 2

Ghosal, S. and van der Vaart, A. (2017). *Fundamentals of nonparametric Bayesian inference*, volume 44 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge. MR3587782. doi: https://doi.org/10.1017/9781139029834. 2

Godsill, S. and Kındap, Y. (2022). "Point process simulation of generalised inverse Gaussian processes and estimation of the Jaeger integral." *Statistics and Computing*, 32(1): Paper No. 13, 18. MR4359569. doi: https://doi.org/10.1007/s11222-021-10072-0. 16, 18

Griffin, J. E. (2016). "An adaptive truncation method for inference in Bayesian

nonparametric models." *Statistics and Computing*, 26(1-2): 423–441. MR3439383. doi: https://doi.org/10.1007/s11222-014-9519-4. 2, 25

Griffin, J. E. and Stephens, D. A. (2013). "Advances in Markov chain Monte Carlo." In *Bayesian theory and applications*, 104–142. Oxford Univ. Press, Oxford. MR3221161. 12

Haario, H., Saksman, E., and Tamminen, J. (2001). "An adaptive Metropolis algorithm." *Bernoulli*, 7(2): 223–242. MR1828504. doi: https://doi.org/10.2307/3318737. 12

Herlau, T., Schmidt, M. N., and Mørup, M. (2016). "Completely random measures for modelling block-structured sparse networks." In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc. 25

Hjort, N. L. (1990). "Nonparametric Bayes estimators based on beta processes in models for life history data." *The Annals of Statistics*, 18(3): 1259–1294. MR1062708. doi: https://doi.org/10.1214/aos/1176347749. 1

Hofert, M. (2011). "Sampling exponentially tilted stable distributions." *ACM Transactions on Modeling and Computer Simulation*, 22(1). MR2955859. doi: https://doi.org/10.1145/2043635.2043638. 24

Horváth, G., Horváth, I., Almousa, S. A.-D., and Telek, M. (2020). "Numerical inverse Laplace transformation using concentrated matrix exponential distributions." *Performance Evaluation*, 137: 102067. MR4104799. doi: https://doi.org/10.1080/15326349.2019.1702058. 19

Ipsen, Y. F. and Maller, R. A. (2017). "Negative binomial construction of random discrete distributions on the infinite simplex." *Theory of Stochastic Processes*, 22(2): 34–46. MR3843523. 6

Ishwaran, H. and James, L. F. (2001). "Gibbs sampling methods for stick-breaking priors." *Journal of the American Statistical Association*, 96(453): 161–173. MR1952729. doi: https://doi.org/10.1198/016214501750332758. 2

Ishwaran, H. and James, L. F. (2002). "Approximate Dirichlet process computing in finite normal mixtures: smoothing and prior information." *Journal of Computational and Graphical Statistics*, 11(3): 508–532. MR1938445. doi: https://doi.org/10.1198/106186002411. 2, 14

James, L. F., Lijoi, A., and Prünster, I. (2009). "Posterior analysis for normalized random measures with independent increments." *Scandinavian Journal of Statistics*, 36(1): 76–97. MR2508332. doi: https://doi.org/10.1111/j.1467-9469.2008.00609.x. 1, 5, 22

Kalli, M., Griffin, J. E., and Walker, S. G. (2011). "Slice sampling mixture models." *Statistics and Computing*, 21(1): 93–105. MR2746606. doi: https://doi.org/10.1007/s11222-009-9150-y. 2, 22, 23

Kim, Y. (1999). "Nonparametric Bayesian estimators for counting processes." *The An-*

nals of Statistics, 27(2): 562–588. MR1714717. doi: https://doi.org/10.1214/aos/1018031207. 1

Kingman, J. F. C. (1967). "Completely random measures." *Pacific Journal of Mathematics*, 21: 59–78. MR0210185. 1

Kingman, J. F. C. (1975). "Random discrete distribution." *Journal of the Royal Statistical Society. Series B. Methodological*, 37: 1–22. MR0368264. 1

Kyprianou, A. E. (2006). *Introductory lectures on fluctuations of Lévy processes with applications*. Universitext. Springer-Verlag, Berlin. MR2250061. 7

Lee, J., James, L. F., and Choi, S. (2016). "Finite-dimensional BFRY priors and variational Bayesian inference for power law models." *Advances in Neural Information Processing Systems*, 29. 2

Lee, J., Miscouridou, X., and Caron, F. (2023). "A unified construction for series representations and finite approximations of completely random measures." *Bernoulli*, 29(3): 2142–2166. MR4580911. doi: https://doi.org/10.3150/22-bej1536. 2, 3

Li, X. and Campbell, T. (2021). "Truncated simulation and inference in edge-exchangeable networks." *Electronic Journal of Statistics*, 15(2): 5117–5157. MR4348694. doi: https://doi.org/10.1214/21-ejs1916. 3, 15

Lijoi, A., Mena, R. H., and Prünster, I. (2005). "Hierarchical mixture modeling with normalized inverse-Gaussian priors." *Journal of the American Statistical Association*, 100(472): 1278–1291. MR2236441. doi: https://doi.org/10.1198/016214505000000132. 1

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. MR2370077. doi: https://doi.org/10.1111/j.1467-9868.2007.00609.x. 1, 16, 21

Muliere, P. and Tardella, L. (1998). "Approximating distributions of random functionals of Ferguson-Dirichlet priors." *The Canadian Journal of Statistics*, 26(2): 283–297. MR1648431. doi: https://doi.org/10.2307/3315511. 2

Naik, C., Caron, F., Rousseau, J., Teh, Y. W., and Palla, K. (2022). "Bayesian Nonparametrics for Sparse Dynamic Networks." In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 191–206. Springer. 3

Neal, R. M. (2000). "Markov chain sampling methods for Dirichlet process mixture models." *Journal of Computational and Graphical Statistics*, 9(2): 249–265. MR1823804. doi: https://doi.org/10.2307/1390653. 2, 22

Neal, R. M. (2011). "MCMC using Hamiltonian dynamics." In *Handbook of Markov chain Monte Carlo*, Chapman & Hall/CRC Handb. Mod. Stat. Methods, 113–162. CRC Press, Boca Raton, FL. MR2858447. 11

Papaspiliopoulos, O. and Roberts, G. O. (2008). "Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models." *Biometrika*, 95(1): 169–186. MR2409721. doi: https://doi.org/10.1093/biomet/asm086. 22

Phadia, E. G. (2016). *Prior processes and their applications*. Springer Series in Statistics. Springer, [Cham], second edition. Nonparametric Bayesian estimation. MR3524072. doi: https://doi.org/10.1007/978-3-319-32789-1. 2

Plummer, M., Best, N., Cowles, K., Vines, K., et al. (2006). "CODA: convergence diagnosis and output analysis for MCMC." *R News*, 6(1): 7–11. 23

Regazzini, E., Lijoi, A., and Prünster, I. (2003). "Distributional results for means of normalized random measures with independent increments." *The Annals of Statistics*, 31(2): 560–585. MR1983542. doi: https://doi.org/10.1214/aos/1051027881. 1, 5

Ricci, F. Z., Guindani, M., and Sudderth, E. (2022). "Thinned random measures for sparse graphs with overlapping communities." In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, 38162–38175. Curran Associates, Inc. 25

Rosiński, J. (2001). "Series representations of Lévy processes from the perspective of point processes." In *Lévy processes*, 401–415. Birkhäuser Boston, Boston, MA. MR1833707. 8, 9, 18

Sokal, A. (1997). "Monte Carlo methods in statistical mechanics: foundations and new algorithms." In *Functional integration (Cargèse, 1996)*, volume 361 of *NATO Adv. Sci. Inst. Ser. B: Phys.*, 131–192. Plenum, New York. MR1477456. 22

Thibaux, R. and Jordan, M. I. (2007). "Hierarchical beta processes and the Indian buffet process." In *Artificial Intelligence and Statistics*, 564–571. PMLR. 1

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. Series B. Statistical Methodology*, 82(2): 487–520. MR4084173. doi: https://doi.org/10.1111/rssb.12363. 25

Walker, S. and Damien, P. (2000). "Representations of Lévy processes without Gaussian components." *Biometrika*, 87(2): 477–483. MR1782492. doi: https://doi.org/10.1093/biomet/87.2.477. 2

Walker, S. G. (2007). "Sampling the Dirichlet mixture model with slices." *Communications in Statistics. Simulation and Computation*, 36(1-3): 45–54. MR2370888. doi: https://doi.org/10.1080/03610910601096262. 2

Williamson, S. A. (2016). "Nonparametric network models for link prediction." *Journal of Machine Learning Research*, 17: Paper No. 202, 21. MR3580355. 5

Wolpert, R. L. and Ickstadt, K. (1998). "Simulation of Lévy random fields." In *Practical nonparametric and semiparametric Bayesian statistics*, volume 133 of *Lect. Notes Stat.*, 227–242. Springer, New York. MR1630084. doi: https://doi.org/10.1007/978-1-4612-1732-9_12. 8

Zhang, J. and Dassios, A. (2023). "Truncated Poisson-Dirichlet approximation for Dirichlet process hierarchical models." *Statistics and Computing*, 33(1): Paper No.

30, 20. MR4530320. doi: https://doi.org/10.1007/s11222-022-10201-3. 3, 4, 18

Zhang, J. and Dassios, A. (2024). Supplementary Material of "Posterior sampling from truncated Ferguson-Klass representation of normalised completely random measure mixtures" doi: https://doi.org/10.1214/24-BA1421SUPP. 16