Slice Sampling for General Completely Random Measures

Peiyuan Zhu **Alexandre Bouchard-Côté** **Trevor Campbell**
Department of Statistics
University of British Columbia
Vancouver, BC V6T 1Z4

Abstract

Completely random measures provide a principled approach to creating flexible unsupervised models, where the number of latent features is infinite and the number of features that influence the data grows with the size of the data set. Due to the infinity the latent features, posterior inference requires either marginalization—resulting in dependence structures that prevent efficient computation via parallelization and conjugacy—or finite truncation, which arbitrarily limits the flexibility of the model. In this paper we present a novel Markov chain Monte Carlo algorithm for posterior inference that adaptively sets the truncation level using auxiliary slice variables, enabling efficient, parallelized computation without sacrificing flexibility. In contrast to past work that achieved this on a model-by-model basis, we provide a general recipe that is applicable to the broad class of completely random measure-based priors. The efficacy of the proposed algorithm is evaluated on several popular nonparametric models, demonstrating a higher effective sample size per second compared to algorithms using marginalization as well as a higher predictive performance compared to models employing fixed truncations.

1 INTRODUCTION

In unsupervised data analysis, one aims to uncover complex latent structure in data. Traditionally, this structure has been assumed to take the form of a *clustering*, in which each data point is associated with exactly one latent category. Here we are concerned with a new generation of unobserved structures such that each data point can be associated to any number of latent categories. When such model can select each category zero or once, the latent categories are called *features* [1], whereas if each category can be selected with multiplicities, the latent categories are called *traits* [2].

Consider for example the problem of modelling movie ratings for a set of users. As a first rough approximation, an analyst may entertain a clustering over the movies and hope to automatically infer movie genres. Clustering in this context is limited; users may like or dislike movies based on many overlapping factors such as genre, actor and score preferences. Feature models, in contrast, support inference of these overlapping movie attributes.

As the amount of data increases, one may hope to capture increasingly sophisticated patterns. We therefore want the model to increase its complexity accordingly. In our movie example, this means uncovering more and more diverse user preference patterns and movie attributes from the growing number of registered users and new movie releases. Bayesian nonparametric methods (BNP) enable unbounded model capacity by positing infinite-dimensional prior distributions. These infinite dimensional priors are designed so that for any given dataset only a finite number of latent parameters are utilized, making Bayesian nonparametric inference possible in practice. The present work is concerned with developing efficient and flexible inference methods for a class of BNP priors called completely random measures (CRMs) [3], which are commonly used in practice [4–7]. In particular, CRMs provide a unified approach to the construction of BNP priors over both latent features and traits [8, 9].

Previous approaches to CRM posterior inference can be categorized into two main types. First, some methods analytically marginalize the infinite dimensional objects involved in CRMs [1, 10–14]. This has the disadvantage of making restrictive conjugacy assumptions and is moreover not amenable to parallelization. A second type of inference method introduced by Blei and Jordan [15] instead uses a fixed truncation of the infinite model. How-

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

ever, this strategy is at odds with the motivation behind BNP, namely, its ability to learn model capacity as part of the inferential procedure. Campbell et al. [16] provide *a priori* error bounds on such truncation, but it is not obvious how to extend these bounds to approximation errors on the posterior distribution.

Our method is based on slice sampling, a family of Markov chain Monte Carlo methods first used in a BNP context by [17]. Slice samplers have advantages over both marginalization and truncation techniques: they do not require conjugacy, enable parallelization, and target the exact nonparametric posterior distribution. But while there is a rich literature on sampling methods for BNP latent feature and trait models—e.g., the Indian buffet / beta-Bernoulli process [13, 18], hierarchies thereof [10], normalized CRMs [12, 19], beta-negative binomial process [14, 20], generalized gamma process [21], gamma-Poisson process [11], and more—these have often been developed on a model-by-model basis.

In contrast to these past model-specific techniques, we develop our sampler based on a *series representation* of the Lévy process [22] underlying the CRM. In a fashion similar to [23] we introduce auxiliary variables that adaptively truncate the series representation; only finitely many latent features are updated in each Gibbs sweep. The representation that we utilize factorizes the weights of CRMs into a transformed Poisson process with independent and identically distributed marks, thereby turning the sampling problem into evaluating the mean measure of a marginalized Poisson point process over a zero-set.

The remainder of the paper is organized as follows: Section 2 introduces the general model that we consider, series representations of CRMs, and posterior inference via marginalization and truncation. Section 3 discusses our main contributions, including model augmentation and slice sampling. Section 4 demonstrates how the methodology can be applied to two popular latent feature models. Finally, in Section 5 we compare our method against several state-of-the-art samplers for these models on both real and synthetic datasets.

2 BACKGROUND

2.1 MODEL

In the standard Bayesian nonparametric latent trait model [16], we are given a data set of random observations $(Y_n)_{n=1}^N$ generated using an infinite collection of latent traits $(\psi_k)_{k=1}^\infty$, $\psi_k \in \Psi$ with corresponding rates $(\theta_k)_{k=1}^\infty$, $\theta_k \in \mathbb{R}_+ := [0, \infty)$. We assume each data point Y_n, $n \in [N] := \{1, \dots, N\}$ is influenced by each trait ψ_k in an amount corresponding to an integer count $X_{nk} \in \mathbb{N}_0 := \{0, 1, 2, \dots\}$ via

$$\begin{aligned} X_{nk} &\overset{\text{indep}}{\sim} h(\cdot; \theta_k) && n, k \in [N] \times \mathbb{N} \\ Y_n &\overset{\text{indep}}{\sim} f\left(\cdot; \sum_{k=1}^\infty X_{nk}\delta_{\psi_k}\right) && n \in [N], \end{aligned}$$

where $\delta_{(\cdot)}$ denotes a Dirac delta measure, h is a distribution on $\mathbb{N}_0$, and f is a distribution on the space of observations. Note that each data point y_n is influenced only by those traits ψ_k for which $x_{nk} > 0$, and the value of x_{nk} denotes the amount of influence.

To generate the infinite collection of (ψ_k, θ_k) pairs, we use a Poisson point process [24] on the product space of traits and rates $\Psi \times \mathbb{R}_+$ with σ-finite mean measure μ,

$$\{\psi_k, \theta_k\}_{k=1}^\infty \sim \mathrm{PP}(\mu) \qquad \mu(\Psi \times \mathbb{R}_+) = \infty. \tag{1}$$

Equivalently, this process can be formulated as a *completely random measure* (CRM) [3] on the space of traits Ψ by placing a Dirac measure at each ψ_k with weight θ_k[1],

$$\sum_{k=1}^\infty \theta_k \delta_{\psi_k} \sim \mathrm{CRM}(\mu). \tag{2}$$

In Bayesian nonparametric modelling, the traits are typically generated independently of the rates, i.e.,

$$\mu\left(\mathrm{d}\theta, \mathrm{d}\psi\right) = \nu\left(\mathrm{d}\theta\right) H\left(\mathrm{d}\psi\right),$$

where H is a probability measure on Ψ, and ν is a σ-finite measure on $\mathbb{R}_+$. In order to guarantee that the CRM has infinitely many atoms, we require that ν satisfies

$$\nu\left(\mathbb{R}_+\right) = \infty,$$

and in order to guarantee that each observation y_n is only influenced by finitely many traits ψ_k having $x_{nk} \neq 0$ a.s., we require that

$$\mathbb{E}\left(\sum_{k=1}^\infty \mathbb{1}\{X_{nk} \neq 0\}\right) = \int (1 - h(0|\theta))\nu\left(d\theta\right) < \infty.$$

To summarize, the model we consider in this paper is:

$$\begin{aligned} &\sum_{k=1}^\infty \theta_k \delta_{\psi_k} \sim \mathrm{CRM}\left(\nu \times H\right) \\ &X_{nk} \overset{\text{indep}}{\sim} h(\cdot; \theta_k) && n, k \in [N] \times \mathbb{N} \\ &Y_n \overset{\text{indep}}{\sim} f\left(\cdot; \sum_k X_{nk}\delta_{\psi_k}\right) && n \in [N]. \end{aligned} \tag{3}$$

[1] More generally, CRMs are the sum of a deterministic measure, an atomic measure with fixed atom locations, and a Poisson point process-based measure as in Eq. (2). In BNP models, there is typically no deterministic component, and the fixed-location atomic component has finitely many atoms, posing no challenge in posterior inference. Thus we focus only on the infinite Poisson point process-based component in this paper.

2.2 SEQUENTIAL REPRESENTATION

While the specification of $\{\psi_k, \theta_k\}_k$ as a Poisson point process is mathematically elegant, it does not lend itself immediately to computation. For this purpose—since there are infinitely many atoms—we require a way of generating them one-by-one in a sequence using familiar finite-dimensional distributions; this is known as a *sequential representation* of the CRM. While there are many such representations (see [16] for an overview), here we will employ the general class of *series representations*, which simulate the traits ψ_k and rates θ_k via

$$E_j \overset{\text{i.i.d.}}{\sim} \mathsf{Exp}(1) \qquad \Gamma_k = \sum_{j=1}^{k} E_j \qquad V_k \overset{\text{i.i.d.}}{\sim} G$$
$$\theta_k = \tau(V_k, \Gamma_k) \qquad \psi_k \overset{\text{i.i.d.}}{\sim} H, \tag{4}$$

where Γ_k are the ordered jumps of a homogeneous, unit-rate Poisson process on $\mathbb{R}_+$, G is a probability distribution on $\mathbb{R}_+$, and $\tau : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ is a nonnegative measurable function such that $\lim_{u\to\infty} \tau(v, u) = 0$ for G-almost every v. For each mean measure μ in Eq. (1), there are many choices of G and τ that together yield a valid series representation for $\mathrm{PP}(\mu)$, such as the inverse-Lévy representation [25], Bondesson representation [26], rejection representation [22], etc.

2.3 POSTERIOR INFERENCE

Posterior inference in the BNP model Eq. (3) is complicated by the presence of infinitely many traits ψ_k and rates θ_k, as the application of traditional MCMC and variational procedures would require infinite computation and memory resources. Past work has handled this issue in two ways: *marginalization* and *fixed truncation*.

Marginalization In a wide variety of CRM-based models, it is possible to analytically integrate out the latent rates and traits [9], thus expressing the model in terms of only the i.i.d. traits ψ_k and sequence of conditional distributions for the assignments X_n,

$$\begin{aligned} \psi_k &\overset{\text{i.i.d.}}{\sim} H && k \in \mathbb{N} \\ X_n &\sim \mathbb{P}\left(X_n = \cdot \,|\, X_{1:n-1}\right) && n \in [N] \\ Y_n &\overset{\text{indep}}{\sim} f\left(\cdot; \sum_k X_{nk}\delta_{\psi_k}\right) && n \in [N]. \end{aligned}$$

Using the exchangeability of the sequence $(X_n)_{n=1}^N$, Gibbs sampling [27] algorithms can be derived that alternate between sampling X_n for each $n \in [N]$, and sampling ψ_k for each of the (finitely many) "active traits" k such that $\sum_n X_{nk} > 0$. However, because each X_n must be sampled conditioned on X_{-n}, these methods cannot be parallelized across n, making them computationally expensive with large amounts of data.

Fixed truncation Another option for posterior inference is to truncate a sequential representation of the CRM such that it generates finitely many traits, i.e.,

$$\begin{aligned} (\psi_k, V_k, \Gamma_k)_{k=1}^K &\sim \text{Eq. (4)} \\ X_{nk} &\overset{\text{indep}}{\sim} h(\cdot; \tau(V_k, \Gamma_k)) && n, k \in [N] \times [K] \\ Y_n &\overset{\text{indep}}{\sim} f\left(\cdot; \sum_{k=1}^{K} X_{nk}\delta_{\psi_k}\right) && n \in [N]. \end{aligned}$$

Because there are only finitely many traits and rates in this model, it is not difficult to develop Gibbs sampling and variational algorithms [28] that iterate between updating the rates $(\theta_k)_{k=1}^K$, the traits $(\psi_k)_{k=1}^K$, and then the assignments $(X_n)_{n=1}^N$. Further, the independence of the assignments across observations n conditioned on the rates and traits enables computationally efficient paralellization of the X update. However, the major drawback of this approach is that the error incurred by truncation is unknown; previous work provides bounds on the total variation distance between the truncated and infinite data marginal distributions [16, 29–31], but error incurred in the posterior distribution is unknown.

3 SLICE SAMPLING FOR CRMs

In this section, we employ an *adaptive* truncation of general CRM series representations to obtain both the computational efficiency of truncated methods and the statistical correctness of approaches based on marginalization. In Section 3.1 we first add an auxiliary variable for each observation n that truncates the full conditional distribution of its underlying assignments X_n. Section 3.2 provides a slice sampling scheme for the augmented model, resulting in truncation that adapts from iteration to iteration.

3.1 AUGMENTED MODEL

We begin by augmenting the model Eq. (3) with auxiliary variables $(U_n)_{n=1}^N$ that truncate the full conditional distributions of the assignments X_n. In particular, suppose we fix the assignments for observations other than n (denoted X_{-n}), the CRM variables ψ, V, Γ, and the auxiliary variables U. Then we require for some $T < \infty$,

$$\forall k > T, \mathbb{P}\left(X_{nk} > 0 \,|\, X_{-n}, \psi, V, \Gamma, U\right) = 0.$$

Past model augmentations in Bayesian nonparametrics have largely required either the normalization of the random measure [12, 17, 19] or a particular sequential representation that guarantees strictly decreasing values of

θ_k [18]. In the present setting of general, unnormalized CRMs, we cannot take advantage of either of these facts.

We therefore take an approach inspired by [23] for augmenting the model. In particular, for each observation $n \in [N]$, define its maximum *active index*

$$k_n := \max \{k \in \mathbb{N} : X_{nk} > 0\} \cup \{0\},$$

and let $\xi : \mathbb{N}_0 \to \mathbb{R}_+$ be a monotone decreasing sequence such that $\lim_{n\to\infty} \xi(n) = 0$. Then we add a uniform random *slice variable* U_n lying in the interval $[0, \xi(k_n)]$ to the model for each observation n, i.e.,

$$\begin{aligned}(\psi_k, V_k, \Gamma_k)_{k=1}^{\infty} &\sim \text{Eq. (4)} \\ X_{nk} &\overset{\text{indep}}{\sim} h(\cdot; \tau(V_k, \Gamma_k)) \qquad n, k \in [N] \times \mathbb{N} \\ U_n &\overset{\text{indep}}{\sim} \mathsf{Unif}\,[0, \xi(k_n)] \qquad n \in [N] \\ Y_n &\overset{\text{indep}}{\sim} f\left(\cdot; \sum_{k=1}^{K} X_{nk}\delta_{\psi_k}\right) \quad n \in [N]. \quad (5)\end{aligned}$$

The variables $(U_n)_{n=1}^N$ do not change the posterior marginal of interest on X, ψ, θ, but do provide computational benefits. In particular, the full conditional distribution of X_n based on Eq. (5) sets $X_{nk} = 0$ for any k such that $\xi(k) < U_n$. Thus, the truncation level for each X_n will adapt as U_n changes from iteration-to-iteration. Further, slice sampling in Eq. (5) requires only finite memory and computation, since we need to store and simulate only those finitely many ψ_k, V_k, Γ_k such that $\xi(k) \geq \min_n U_n$ at each iteration. The ability to instantiate the latent ψ_k, V_k, Γ_k variables has many advantages; e.g., we can leverage the independence of X_n for parallelization without sacrificing the fidelity of the model. The augmented probablistic model is depicted in Fig. 1.

3.2 SLICE SAMPLING

In this section, we develop a slice sampling scheme for the augmented model Eq. (5) that iteratively simulates from each full conditional distribution. The state of the Markov chain that we construct is infinite-dimensional, consisting of $(X_{nk})_{n\in[N],k\in\mathbb{N}}$, $(\psi_k, V_k, \Gamma_k)_{k\in\mathbb{N}}$, and $(U_n)_{n\in[N]}$. Due to the augmentation in Eq. (5), however, only finitely many of these variables need to be stored or simulated during any iteration of the algorithm. The particular steps follow; note that the order of the steps is important.

Initialization Set the assignment variables $X = 0$, the global truncation levels $K = K_{\text{prev}} = 0$, and for all $n \in [N]$, the local truncation levels $k_n = k'_n = 0$. Run this step only a single time at the beginning of the algorithm.

Sample U: For $n \in [N]$, draw $U_n \overset{\text{indep}}{\sim} \mathsf{Unif}\,[0, \xi(k_n)]$.

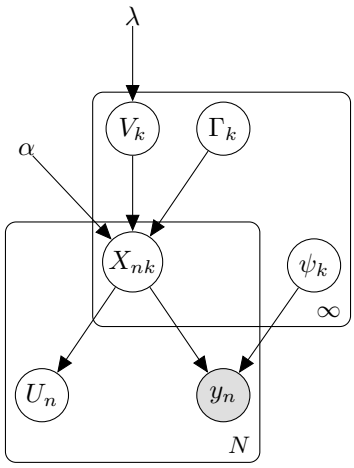


Figure 1: Probablistic graphical model based on series representation of CRM in plate notation after augmentation with auxiliary variables U.

Update global truncation level: Set

$$\begin{aligned}K_{\text{prev}} &\leftarrow \max_{n\in[N]} k_n \\ K &\leftarrow \max\left\{k \in \mathbb{N}_0 : \xi(k) \geq \min_n U_n\right\}.\end{aligned}$$

In other words, K is the maximum index that observations might activate in this iteration, and K_{prev} is the maximum index that observations activated in the previous iteration. Note that Γ_k, V_k, ψ_k are all guaranteed to be instantiated for $k = 1, \ldots, K_{\text{prev}}$.

Sample ψ For each $k \in [K]$, sample ψ_k from its full conditional distribution, with measure proportional to

$$H(\mathrm{d}\psi_k) \prod_{n=1}^{N} f\left(Y_n; X_{nk}\delta_{\psi_k} + \sum_{j\neq k} X_{nj}\delta_{\psi_j}\right).$$

If possible, $(\psi_k)_{k=1}^K$ should instead be sampled jointly from the same density above. Further, if f is not conjugate to the prior measure H, this step may be conducted using Metropolis-Hastings. Since the remaining values $(\psi_k)_{k=K+1}^{\infty}$ will not influence the remainder of this iteration, they do not need to be simulated.

Sample V, Γ This step is split into two substeps: first sample $(V_k, \Gamma_k)_{k=1}^{K_{\text{prev}}-1}$ each from their own full conditional, and then sample $(V_k, \Gamma_k)_{k=K_{\text{prev}}}^{\infty}$ as a single block. Define $\Gamma_0 := 0$ for notational convenience.

Substep 1: Note that V_k, Γ_k are conditionally independent of all other variables given $(X_{nk})_{n=1}^N$, Γ_{k-1}, and Γ_{k+1}. Thus, for each $k \in [K_{\text{prev}}]$, we generate each pair

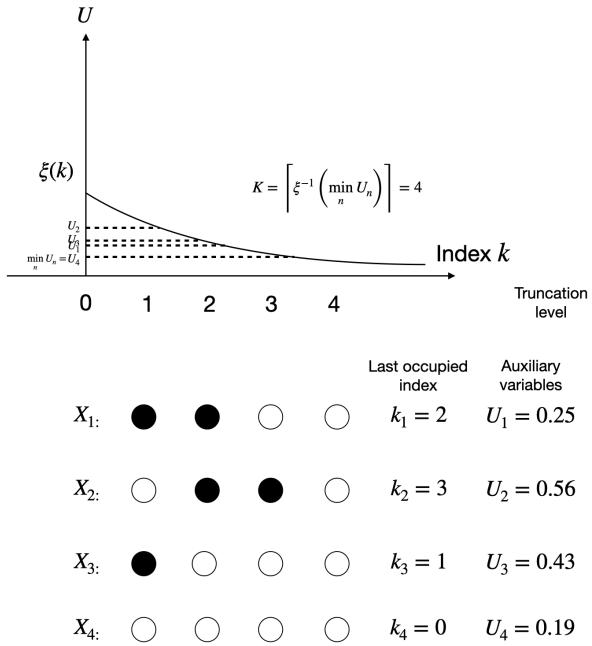


Figure 2: An instance of slice variables in a dataset with four observations.

of V_k, Γ_k from its full conditional, with measure proportional to

$$G(\mathrm{d}V_k)\mathrm{d}\Gamma_k \mathbb{1}\left[\Gamma_{k-1} \leq \Gamma_k \leq \Gamma_{k+1}\right] \cdot \prod_{n=1}^{N} h(X_{nk}; \tau(V_k, \Gamma_k)).$$

Substep 2: If $K_{\text{prev}} > K$, this step is skipped. Otherwise, for each $k \in \{K_{\text{prev}}, \ldots, K\}$ in increasing order, we generate V_k, Γ_k conditioned on $V_{1:k-1}, \Gamma_{1:k-1}$ and the remaining variables. In particular, note that V_k, Γ_k are conditionally independent of all other variables given $(X_{nj})_{n\in[N],j\geq k}$ and Γ_{k-1}. Thus, we generate V_k, Γ_k from a measure proportional to

$$G(\mathrm{d}V_k)\mathrm{d}\Gamma_k \exp\left(-\Gamma_k\right) \mathbb{1}\left[\Gamma_k \geq \Gamma_{k-1}\right] \cdot \mathbb{P}\left(\left(X_{nj}\right)_{n\in[N],j>k} = 0 | \Gamma_k\right) \cdot \prod_{n=1}^{N} h\left(X_{nk}; \tau(V_k, \Gamma_k)\right)$$

Again, since the remaining values $(V_k, \Gamma_k)_{k=K+1}^{\infty}$ will not influence the remainder of this iteration, they do not need to be simulated and can be safely ignored.

In order to evaluate $\mathbb{P}\left(\left(X_{nj}\right)_{n\in[N],j>k} = 0 \,|\, \Gamma_k\right)$, we use the machinery of Poisson point processes. In particular, note that the independent generation of X_{nk} given V_k, Γ_k ensures that $\Pi := \{\Gamma_j, V_j, X_{nj}\}_{n\in[N],j>k}$ conditioned on Γ_k is itself a Poisson point process on the joint space by repeated use of the marking theorem [24, Ch. 5.2]. Therefore, this probability can be written as the probability that Π has no atoms with a nonzero integer component:

$$\begin{aligned}&\mathbb{P}\left(X_{n\in[N],j>k} = 0 \,|\, \Gamma_k\right) \\ =&\mathbb{P}\left(\left|\Pi \cap \left(\mathbb{R}^2 \times \mathbb{N}^N\right)\right| = 0 \,|\, \Gamma_k\right),\end{aligned}$$

which itself can be written explicitly using the fact that the number of atoms of a Poisson point process in any set has a Poisson distribution:

$$\begin{aligned}&= \exp\left(-\int_{\gamma\geq\Gamma_k} \sum_{x\in\mathbb{N}^N} h(x \,|\, \tau(v,\gamma)) G(\mathrm{d}v)\mathrm{d}\gamma\right) \\ &= \exp\left(-\int_{\gamma\geq\Gamma_k} \left(1 - h\left(0 \,|\, \tau(v,\gamma)\right)^N\right) G(\mathrm{d}v)\mathrm{d}\gamma\right) \quad (6)\end{aligned}$$

If this expression cannot be evaluated exactly but an upper bound is available, then rejection sampling may be used. It is worth noting that Eq. (6) appears in past bounds on the incurred error by truncating completely random measures [16, Eqn 4.2]; here we bridge the connection between truncation and slice sampling; if a tight bound of the truncation error exists, then efficient rejection sampling scheme can be developed accordingly. Prior to inference, the integral Eq. (6) can be precomputed for a range of Γ_ks; afterward, it can be evaluated via interpolation.

Discard unused traits Discard ψ_k, V_k, Γ_k for $k > K$; this step will only occur if $K_{\text{prev}} > K$.

Sample X For each $n \in [N]$ and $k \in [K]$, note that X_{nk} is conditionally independent of all other variables given $V_k, \Gamma_k, U_n, (X_{nj})_{j\neq k}$. Simulate the integer value of X_{nk} from its full conditional distribution with measure proportional to

$$f\left(Y_n; X_{nk}\delta_{\psi_k} + \sum_{j\neq k} X_{nj}\delta_{\psi_j}\right) h\left(X_{nk} | \tau\left(V_k, \Gamma_k\right)\right) \cdot \xi\left(\hat{k}_n(X_{nk})\right)^{-1} \mathbb{1}\left[U_n \leq \xi\left(\hat{k}_n\left(X_{nk}\right)\right)\right],$$

where the function $\hat{k}_n : \mathbb{N}_0 \to \mathbb{N}_0$ is defined by

$$\hat{k}_n(x) := \begin{cases} k'_n & x = 0, k = k_n \\ k & x > 0, k > k_n \\ k_n & \text{otherwise} \end{cases},$$

i.e., it computes what k_n *would be* as we vary the value of X_{nk}. Note that this step can be parallelized across the N data points. Further, note that $(X_{nk})_{n\in[N],k>K}$ are all guaranteed to be 0 due to the truncation at level K, and so do not need to be simulated. If the density is intractable, we can use Metropolis-Hasting within Gibbs for this step.

Algorithm 1 Slice sampling for general CRMs

procedure SLICE SAMPLER(y, M K, f, h, τ, G)
  Initialize $X, K_{\text{prev}}, K, k, k'$
  for $m \leftarrow 1, \cdots, M$ **do**:
    Sample U_n for $n \in [N]$
    Update K, K_{prev}
    Resize X, ψ, Γ, V to K
    Sample ψ_k for $k \in [K]$
    for $k \leftarrow 1, \cdots, K$ **do**
      if $k < K_{\text{prev}}$ **then**:
        Sample $\Gamma_k, V_k | \Gamma_{k-1}, \Gamma_{k+1}$
      else:
        Sample $\Gamma_k, V_k | \Gamma_{k-1}$
      end if
    end for
    Sample X_{nk} for $n \in [N], k \in [K]$
    Update k, k'
  end for
end procedure

Update local truncation levels For each $n \in [N]$, compute the first and second maximum active indices,

$$k_n \leftarrow \max\{k \in \mathbb{N} : X_{nk} > 0\} \cup \{0\}$$
$$k'_n \leftarrow \max\{k < k_n : X_{nk} > 0\} \cup \{0\}.$$

Most probablistic graphical models with CRM component can be converted to the form of 3. This algorithm then takes the input of a probablistic graphical model with samplers that sample from the conditional distributions and pointers to the variables Y, X, and ψ. This procedure is shown in pseudocode in Algorithm 1.

4 APPLICATIONS

In this section, we show how the general slice sampler from Section 3 can be applied to two popular Bayesian nonparametric models: the beta-Bernoulli latent feature model [1, 32] and beta-negative binomial (BNB) combinatorial clustering model [20]. In particular, we provide the details of each of the sampling steps from Section 3.2 based on the Bondesson series representation [26] of the beta process. The beta process with mass parameter α and shape parameter λ has rate measure

$$\nu(d\theta) = \lambda\alpha\theta^{-1}(1-\theta)^{\lambda-1}d\theta,$$

and has a Bondesson representation [16, 26] for the rates

$$\theta_k = V_k \exp\left(-\Gamma_k/(\lambda\alpha)\right)$$
$$(\Gamma_k)_{k=1}^{\infty} \sim \text{PP}(1), \quad V_k \overset{\text{i.i.d.}}{\sim} \mathsf{Beta}(1, \lambda - 1),$$

where $\text{PP}(1)$ is shorthand for a unit-rate homogeneous Poisson process on $[0, \infty)$. The beta process can be paired with a Bernoulli likelihood

$$h(x|\theta) = \theta^x (1-\theta)^{1-x}, \quad x \in \{0, 1\},$$

or negative binomial likelihood,

$$h(x|\theta, r) = \binom{x+r-1}{x} (1-\theta)^r \theta^x, \quad x \in \mathbb{N} \cup \{0\}.$$

In both of the following examples, we set the monotone sequence $\xi(k)$ to $\xi(k) = \exp(-k/\Delta_\xi)$, where $\Delta_\xi > 0$ is a hyperparameter to be tuned in each case.

4.1 BETA-BERNOULLI FEATURE MODEL

Given a dataset of observations $y_n \in \mathbb{R}^d$, $n = 1, \ldots, N$, the beta-Bernoulli latent feature model aims to uncover a collection of latent features $\psi_k \in \mathbb{R}^d$ and binary assignments $X_{nk} \in \{0, 1\}$ of data to features responsible for generating the observations:

$$\{\Gamma_k\}_{k=1}^{\infty} \sim \text{PP}(1)$$
$$X_{nk} \overset{\text{indep}}{\sim} \mathsf{Bern}\left(\exp\left(-\frac{\Gamma_k}{c}\right)\right), \quad n \in [N]$$
$$\psi_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma_0^2 I\right), \quad k \in \mathbb{N}$$
$$y_n \overset{\text{indep}}{\sim} \mathcal{N}\left(\sum_k X_{nk}\psi_k, \sigma^2 I\right), \quad n \in [N].$$

We assume the hyperparameters σ_0^2, σ^2, and c are given. We don't need to sample V because here $G = \delta_1$.

Sample X: For each $n \in [N]$ and $k \in [K]$, we sample $X_{nk} = r \in \{0, 1\}$ with probability proportional to

$$\mathcal{N}\left(y_n; \sum_{j=1, j\neq k}^{\infty} X_{nj}\psi_j + r\psi_k, \sigma^2 I\right) \cdot$$
$$\mathsf{Bern}\left(r; \exp\left(-\frac{\Gamma_k}{c}\right)\right) \mathsf{Unif}\left(U_n; 0, \xi\left(\hat{k}_n(r)\right)\right),$$

where $\mathcal{N}(\cdot; \ldots)$, $\mathsf{Bern}(\cdot; \ldots)$, and $\mathsf{Unif}(\cdot; \ldots)$ are the density functions of the respective distributions. One may parallelize this sampling step across $n \in [N]$ due to the introduction of auxiliary variables.

Sample ψ: Using the conjugacy of the feature prior and data likelihood, we sample all the features $(\psi_k)_{k=1}^K$ simultaneously,

$$Q = X^T X + \frac{\sigma^2}{\sigma_0^2} I$$
$$\psi \sim \mathcal{MN}\left(\psi; Q^{-1}\left(X^T y\right), Q^{-1}, \sigma^2 I\right),$$

where $\mathcal{MN}(\cdot; \ldots)$ is the density function for the matrix normal distribution [33].

Sample Γ (substep 1): The full conditional distribution of Γ_k has density proportional to

$$e^{-m_k\Gamma_k/c}\left(1-e^{-\Gamma_k/c}\right)^{N-m_k}\mathbb{1}\left[\Gamma_{k-1}\leq\Gamma_k\leq\Gamma_{k+1}\right],$$

where $m_k := \sum_{n=1}^{N} X_{nk}$. Rather than simulating from this density exactly—which would require expensive iterative numerical integration—we use Metropolis-Hastings to sample Γ_k, with proposal

$$\begin{aligned} W &\sim \mathsf{Unif}\left[-\Delta_\Gamma,\Delta_\Gamma\right] \\ \Gamma_k' &= W+\max\left(\min\left(\Gamma_k,\Gamma_{k+1}-\Delta_\Gamma\right),\Gamma_{k-1}+\Delta_\Gamma\right) \end{aligned} \tag{7}$$

for step size $\Delta_\Gamma > 0$ and $\Delta_\Gamma = \frac{\Gamma_{k+1}-\Gamma_{k-1}}{n_\Gamma}$ by dividing the interval length into n_Γ pieces.

Sample Γ (substep 2): The expansion distribution of Γ_k has density proportional to

$$e^{-(1+m_k/c)\Gamma_k-I(\Gamma_k)}\left(1-e^{-\Gamma_k/c}\right)^{N-m_k}\mathbb{1}\left[\Gamma_k\geq\Gamma_{k-1}\right]$$

where $m_k := \sum_{n=1}^{N} X_{nk}$, and

$$I\left(\Gamma_k\right)=\int_{\Gamma_k}^{\infty}1-\exp\left(-Ne^{-\gamma/c}\right)d\gamma.$$

We again use Metropolis-Hastings to sample and Γ_k. For convenience we set $\Delta_\Gamma = \frac{1}{n_\Gamma}$ for this step in particular.

Note that $I(\gamma)$ can be precomputed using numerical integration at a wide range of points prior to slice sampling; here we chose to precompute $I(x)$ at 1000 evenly spaced points $e^{-x/c} \in [\epsilon, 1]$ for $\epsilon = 10^{-30}$. During MCMC, we evaluate $I(\Gamma_k)$ with spline-interpolation.

4.2 BNB CLUSTERING MODEL

Given a collection of documents $d = 1,\ldots,D$ each containing $N_d \in \mathbb{N}$ words $y_{dn} \in [W]$, $W \in \mathbb{N}$, BNB combinatorial clustering aims to uncover latent topics $\psi_k \in [0,1]^V$, $\sum_w \psi_{kw} = 1$, and document-specific topic rates $\pi_{dk} > 0$, $k = 1,\ldots,\infty$, via

$$\begin{aligned} \{\Gamma_k\}_{k=1}^{\infty} &\sim \mathrm{PP}\left(1\right) \\ V_k &\overset{\text{i.i.d.}}{\sim} \mathsf{Beta}\left(1,\lambda-1\right) \\ \theta_{dk} &\overset{\text{indep}}{\sim} \mathsf{Beta}\left(\alpha\lambda V_k e^{-\frac{\Gamma_k}{c}},\lambda\left(1-\alpha V_k e^{-\frac{\Gamma_k}{c}}\right)\right) \\ \psi_k &\overset{\text{i.i.d.}}{\sim} \mathsf{Dir}\left(\beta\right) \\ \pi_{dk} &\overset{\text{indep}}{\sim} \mathsf{Gam}\left(r,\frac{1-\theta_{dk}}{\theta_{dk}}\right) \\ Z_{dn} &\overset{\text{indep}}{\sim} \mathsf{Categorical}\left(\left(\frac{\pi_{dk}}{\sum_k \pi_{dk}}\right)_{k=1}^{\infty}\right) \\ y_{dn} &\overset{\text{indep}}{\sim} \mathsf{Categorical}\left(\psi_{z_{dn}}\right). \end{aligned}$$

Here Z_{dn} is the topic indicator of word n in document d. We assume the hyperparameters $\lambda > 1$, $0 < \alpha < 1$ and set $c = \lambda\alpha$. $\beta \in \mathbb{R}_+^V$, and $r > 0$ are given. In this model, note that we sample per-word auxiliary variables:

$$U_{dn} \sim \mathsf{Unif}\left[0,\xi\left(Z_{dn}\right)\right].$$

Sample Z The conditional distribution of Z_{dn} is a categorical thresholded by the auxiliary variable; in particular, the probability that $Z_{dn} = k$ is proportional to

$$\pi_{dk}\psi_{ky_{dn}}\mathbb{1}\left[U_{dn}\leq\xi\left(k\right)\right]/\xi(k).$$

Sample ψ We sample the latent features exactly from their full conditional Dirichlet distribution via

$$\psi_k \sim \mathsf{Dir}\left(\left(\beta_v+\sum_{(n,d):z_{dn}=k}\mathbb{1}\{y_{dn}=w\}\right)_{w=1}^{W}\right).$$

The calculation is standard and similar to [34].

Sample V,Γ (substep 1): The full conditional distribution of V_k, Γ_k has density proportional to

$$\begin{aligned} &\prod_{d=1}^{D}\mathsf{BNB}\left(X_{dk};r,\alpha\lambda V_k e^{-\frac{\Gamma_k}{c}},\lambda\left(1-\alpha V_k e^{-\frac{\Gamma_k}{c}}\right)\right) \\ &\cdot\,\mathsf{Beta}\left(V_k;1,\lambda-1\right)\mathsf{Unif}\left[\Gamma_k;\Gamma_{k-1},\Gamma_{k+1}\right] \end{aligned}$$

where $X_{dk} = \sum_{n=1}^{N_d}\mathbb{1}\left[Z_{nd}=k\right]$, and $\mathsf{BNB}(\cdot;\ldots)$, $\mathsf{Beta}(\cdot;\ldots)$, and $\mathsf{Unif}(\cdot;\ldots)$ are the density functions for the beta-negative binomial, beta, and uniform distributions. Here the X_{dk}'s are conditionally independent because we do not condition on a fixed number of words in each document N_d.

We again use Metropolis-Hastings to sample V_k and Γ_k. Here V is sampled by a random walk proposal for with hyperparameter $\triangle_V$.

$$\begin{aligned} E &\sim \mathsf{Unif}\left[-\Delta_V,\Delta_V\right] \\ V_k' &= E+\max\left(\min\left(V_k,1-\Delta_V\right),\Delta_V\right), \end{aligned}$$

and Γ is sampled using the same algorihtm as 7.

Sample V,Γ (substep 2): The expansion distribution of V_k, Γ_k has density proportional to

$$\begin{aligned} &\exp\left(-I\left(\Gamma_k\right)\right) \\ &\prod_{d=1}^{D}\mathsf{BNB}\left(X_{dk};r,\alpha\lambda V_k e^{-\frac{\Gamma_k}{c}},\lambda\left(1-\alpha V_k e^{-\frac{\Gamma_k}{c}}\right)\right) \\ &\mathsf{Beta}\left(V_k;1,\lambda-1\right)\mathsf{Exp}\left(\Gamma_k-\Gamma_{k-1};1\right), \end{aligned}$$

where the integral expression is given by

$$I(\Gamma_k) = \int_{\Gamma_k}^{\infty} \int_0^1 (1 - F(v, \gamma)) \, \mathsf{Beta}(v; 1, \lambda - 1) \, dv d\gamma$$

$$F(v, \gamma) = \mathsf{BNB}\left(0; r, \alpha\lambda v e^{-\frac{\gamma}{c}}, \lambda\left(1 - \alpha v e^{-\frac{\gamma}{c}}\right)\right)^D.$$

For simplicity the formula presented here is for beta-negative binomial likelihood with homogeneous failure probability but we set them differently in later experiments. This density can be jointly sampled using Metropolis-Hastings with thresholded uniform proposals, similarly to the previous substep. Furthermore, as in substep 2 of the previous example, this integral can be precomputed for a range of values before sampling.

5 EXPERIMENTS

In this section, we compare the performance of our algorithm on the two models described in Sections 4.1 and 4.2. On both synthetic and real datasets our algorithm outperforms state-of-the-art methods and fixed truncation.

5.1 BETA-BERNOULLI FEATURE MODEL

In the first experiment we generate synthetic data from a truncated version of the beta-Bernoulli model from Section 4.1, with model parameters set to $(\sigma, \sigma_0, c) = (0.2, 0.5, 1)$. We test a number of experimental settings: for each $N \in \{10000, 11000, \cdots, 19000, 20000\}$, we set the synthetic generating model truncation level to $K = 2\lceil \log(N) \rceil$ and data dimension $D = 2\lceil \frac{N \log(N)}{N - \log(N)} \rceil$, such that the features are roughly identifiable from the data. For the auxiliary variables in the slice sampler, we set the scale of the ξ sequence to $\Delta_\xi = 1$. The scale is optimized over $\{0.1, 0.2, 0.3, 0.4, ..., 2.9, 3\}$ to maximize effective sample size per second (ESS/s) using simulated data (N=1000). We set the Metropolis-Hastings step size to $n_\Gamma = 10$. This chosen from the set $1, 2, \ldots, 10$ to result in a MH acceptance rate of Γ averaged over iterations to be roughly between 0.2 and 0.9, which is a standard general practice in MCMC methods. We generate synthetic data and run the proposed slice sampling algorithm for 1,000 iterations over 10 independent trials, comparing to the state-of-the-art accelerated collapsed Gibbs sampler by Doshi-Velez and Ghahramani [13] with $\mathcal{O}(N^2)$ runtime per MCMC iteration. We perform the comparison by measuring both ESS/s and 2-norm error of held-out data. The error is evaluated with latent features from the Monte Carlo samples and combinatorial variable X chosen to minimize error. $(N_{train}, N_{test}, K, c, \sigma, \sigma_0, \triangle_\xi) = (300, 200, 20, 2, 0.5, 0.5)$ where the parameters that require tuning are tuned in a procedure similar to the previous experiment. To evaluate the ESS/s for both samplers, we compute a test function that returns 1 if the combinatorial matrix X has an even number of non-zero entries and 0 otherwise, and use the batch mean estimator [35].

The results are shown in Figs. 3a and 3b. Fig. 3a suggests our sampler has ESS/s that scales as $\mathcal{O}(N^{-0.6})$, while the accelerated collapsed sampler has ESS/s that scales as $\mathcal{O}(N^{-1.34})$; this improvement arises from the linear runtime per-iteration compared to the quadratic runtime of the algorithm of [13]. Fig. 3b achieved the smallest error by quickly selecting a suitable truncation level.

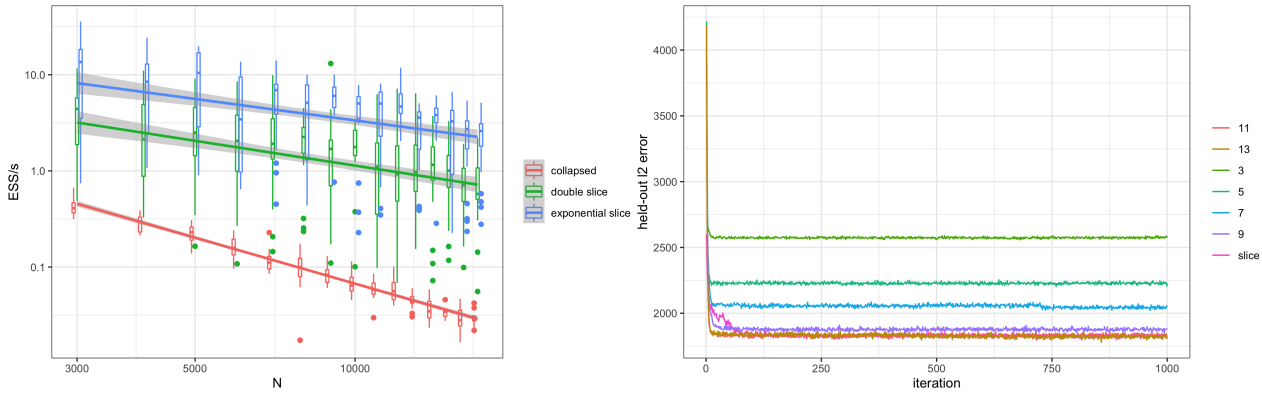
5.2 HIERARCHICAL CLUSTERING

In the second experiment, we used the BNB clustering model from Section 4.2 to analyze the NeurIPS corpus from 2010 to 2015, preprocessed to remove stopwords and truncate the vocabulary to those words appearing more than 50 times. We randomly split each document of the NeurIPS papers corpus into held-out test words (30%) and training words (70%). We set concentration and scale parameters to $(\alpha, \lambda) = (1, 1.1)$, the prior Dirichlet topic distribution parameter over the V vocabulary words to $\beta = (0.1, \cdots, 0.1)$, and the failure rate of the negative binomial distribution is set to $r_d = \frac{N_d(\lambda - 1)}{\alpha\lambda}$ for each document $d \in \{1, \ldots, D\}$. We set the Metropolis-Hastings step sizes to $(n_\Gamma, \Delta_V) = (10, 0.3)$, and the scale of the ξ sequence to $\Delta_\xi = 3$. These parameters are tuned similar to the previous section. We compared our slice sampler to the slice sampler of [20] on both ESS/s and held-out data perplexity. We use the same procedure as in the previous experiment to estimate ESS/s.

The results are shown in Figs. 4a and 4b. Fig. 4a shows that our algorithm produces a roughly two orders of magnitude improvement in ESS/s on large datasets than the comparison method. Fig. 4b demonstrates that the proposed slice sampler also provides a significant decrease in the held-out test set perplexity [34]. This is at least in part because the proposed slice sampler is generic and can use any series representation of the underlying CRM; here, we take advantage of that and use the Bondesson representation, which is known to provide exponentially decreasing truncation error [16] and is significantly more efficient than the superposition representation used by [20]. In practice, this manifests as a high number of unused or redundant atoms in past samplers, while the proposed sampler does not exhibit this issue.

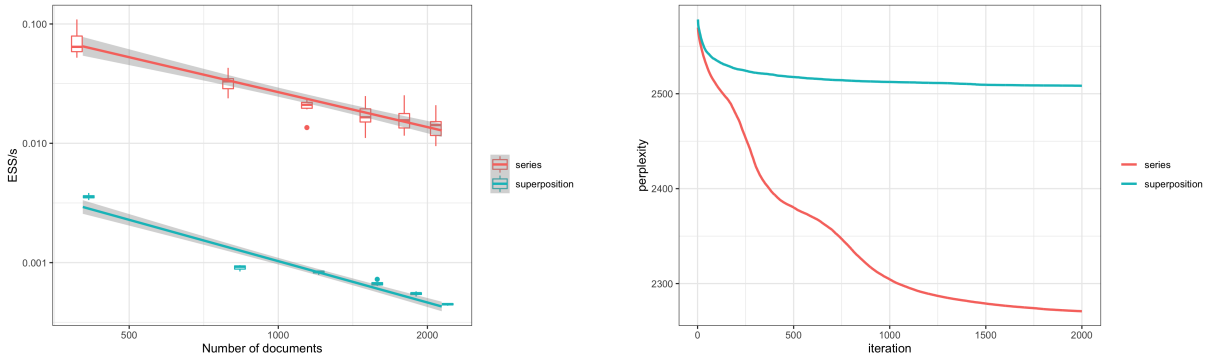
6 CONCLUSION

In this paper, we introduced a computational method for posterior inference in a large class of unsupervised Bayesian nonparametric models. Compared with past



(a) (b)

Figure 3: (3a): Boxplot and linear fit of the ESS/s of 20 runs of model. Both ESS/s and N are plotted in log scale of base 10. The line "double slice" corresponds to auxiliary variable $U_n \equiv K_n \sim \mathsf{Unif}\,[k'_n, 2k'_n]$. (3b): 2-norm error of held-out data for truncated and adpatively truncated model (slice sampler) optimized over the combinatorial space.



(a) (b)

Figure 4: (4a): Boxplot and linear fit of the ESS/s of 6 runs of model. Both ESS/s and N are plotted in log scale of base 10. (4b): Perplexity evaluated on the test set at each iteration.

work, our method enables parallel inference, does not require conjugacy, and targets the exact posterior.

It is worth noting that the proposed sampler does not necessarily generalize past slice samplers for specific models (e.g., [18]). Although model-specific methods may provide performance gains in some cases, our method can be easily incorporated in a general probabilistic programming system, and provides parallel inference for a wide range of models. Future work on the proposed methodology could include automated selection of the deterministic sequence ξ, and the incorporation of more advanced Markov chain moves, such as split-merge [36].

Acknowledgements

This research is supported by a National Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant and Discovery Launch Supplement.

References

[1] Thomas Griffiths and Zoubin Ghahramani. Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems*, 2005.

[2] Trevor Campbell, Diana Cai, and Tamara Broderick. Exchangeable trait allocations. *Electronic Journal of Statistics*, 12:2290–2322, 2018.

[3] John Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.

[4] Diana Cai and Tamara Broderick. Completely random measures for modeling power laws in sparse graphs. *NeurIPS Workshop on Networks in the Social and Information Sciences*, 2015.

[5] Sunil Gupta, Dinh Phung, and Svetha Venkatesh. Factorial multi-task learning: A Bayesian nonparametric approach. In *International Conference on Machine Learning*, 2013.

[6] John Paisley, David Blei, and Michael Jordan. Stick-breaking beta processes and the Poisson process. In *Artificial Intelligence and Statistics*, 2012.

[7] Ayan Acharya, Joydeep Ghosh, and Mingyuan Zhou. Nonparametric Bayesian factor analysis for dynamic count matrices. In *Artificial Intelligence and Statistics*, 2015.

[8] Michael Jordan. Hierarchical models, nested models and completely random measures. *Frontiers of statistical decision making and Bayesian analysis: In honor of James O. Berger. New York: Springer*, pages 207–218, 2010.

[9] Tamara Broderick, Ashia Wilson, and Michael Jordan. Posteriors, conjugacy, and exponential families for completely random measures. *Bernoulli*, 24(4):3181–3221, 2018.

[10] Romain Thibaux and Michael Jordan. Hierarchical Beta processes and the Indian buffet process. In *Artificial Intelligence and Statistics*, 2007.

[11] Michalis Titsias. The infinite gamma-Poisson feature model. In *Advances in Neural Information Processing Systems*, 2008.

[12] Jim Griffin and Stephen Walker. Posterior simulation of normalized random measure mixtures. *Journal of Computational and Graphical Statistics*, 20(1):241–259, 2011.

[13] Finale Doshi-Velez and Zoubin Ghahramani. Accelerated sampling for the Indian buffet process. In *International Conference of Machine Learning*, 2009.

[14] Mingyuan Zhou, Lauren Hannah, David Dunson, and Lawrence Carin. Beta-negative binomial process and Poisson factor analysis. In *Artificial Intelligence and Statistics*, 2012.

[15] David Blei and Michael Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1): 121–144, 2006.

[16] Trevor Campbell, Jonathan Huggins, Jonathan How, and Tamara Broderick. Truncated random measures. *Bernoulli*, 25(2):1256–1288, 2019.

[17] Stephen Walker. Sampling the Dirichlet mixture model with slices. *Communications in Statistics - Simulation and Computation*, 36(1):45–54, 2007.

[18] Yee Whye Teh, Dilan Görür, and Zoubin Ghahramani. Stick-breaking construction for the Indian buffet process. In *Artificial Intelligence and Statistics*, 2007.

[19] Stefano Favaro and Yee Whye Teh. MCMC for normalized random measure mixture models. *Statistical Science*, 28 (3):335–359, 2013.

[20] T. Broderick, L. Mackey, J. Paisley, and M. Jordan. Combinatorial clustering and the beta negative binomial process. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):290–306, 2015.

[21] Fadhel Ayed and François Caron. Nonnegative Bayesian nonparametric factor models with completely random measures for community detection. *arXiv:1902.10693*, 2019.

[22] Jan Rosiński. Series representations of Lévy processes from the perspective of point processes. In *Lévy Processes: Theory and Applications*, pages 401–415. Springer, 2001.

[23] Maria Kalli, Jim Griffin, and Stephen Walker. Slice sampling mixture models. *Statistics and Computing*, 21(1): 93–105, 2011.

[24] John Kingman. *Poisson Processes*. Clarendon Press, 1992.

[25] Thomas Ferguson and Michael Klass. A representation of independent increment processes without gaussian components. *The Annals of Mathematical Statistics*, 43(5): 1634–1643, 1972.

[26] Lennart Bondesson. On simulation from infinitely divisible distributions. *Advances in Applied Probability*, 14: 855–869, 1982.

[27] Martin Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987.

[28] Anirban Roychowdhury and Brian Kulis. Gamma processes, stick-breaking, and variational inference. In *Artificial Intelligence and Statistics*, 2015.

[29] Julyan Arbel and Igor Prünster. A moment-matching Ferguson & Klass algorithm. *Statistics and Computing*, 27 (1):3–17, 2017.

[30] Raffaele Argiento, Ilaria Bianchini, and Alessandra Guglielmi. A blocked Gibbs sampler for NGG-mixture models via a priori truncation. *Statistics and Computing*, 26(3):641–661, 2016.

[31] Finale Doshi, Kurt Miller, Jurgen Van Gael, and Yee Whye Teh. Variational inference for the Indian buffet process. In *Artificial Intelligence and Statistics*, 2009.

[32] Nils Lid Hjort. Nonparametric Bayes estimators based on beta processes in models for life history data. *The Annals of Statistics*, 18(3):1259–1294, 1990.

[33] Philip Dawid. Some matrix-variate distribution theory: Notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274, 1981.

[34] David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[35] James Flegal and Galin Jones. Batch means and spectral variance estimators in Markov chain Monte Carlo. *The Annals of Statistics*, 38(2):1034–1070, 2010.

[36] Sonia Jain and Radford Neal. Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Analysis*, 2(3), 2007.