Received: 14 November 2022 | Revised: 23 July 2023 | Accepted: 12 September 2023

DOI: 10.1111/sjos.12688

ORIGINAL ARTICLE

Scandinavian Journal of Statistics

Truncated two-parameter Poisson–Dirichlet approximation for Pitman–Yor process hierarchical models

Junyi Zhang[1,2] | **Angelos Dassios**[1]

[1]Department of Statistics, London School of Economics, London, UK

[2]Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong

Correspondence
Junyi Zhang, Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong.
Email: JunyZhang@polyu.edu.hk

Abstract

In this paper, we construct an approximation to the Pitman–Yor process by truncating its two-parameter Poisson–Dirichlet representation. The truncation is based on a decreasing sequence of random weights, thus having a lower approximation error compared to the popular truncated stick-breaking process. We develop an exact simulation algorithm to sample from the approximation process and provide an alternative MCMC algorithm for the parameter regime where the exact simulation algorithm becomes slow. The effectiveness of the simulation algorithms is demonstrated by the estimation of the functionals of a Pitman–Yor process. Then we adapt the approximation process into a Pitman–Yor process mixture model and devise a blocked Gibbs sampler for posterior inference.

KEYWORDS

Bayesian nonparametric statistics, Markov chain Monte Carlo, mixture model, Pitman–Yor process, two-parameter Poisson–Dirichlet distribution

1 | INTRODUCTION

The Pitman–Yor process is a rich and flexible class of random probability measures that has been widely used in Bayesian nonparametric statistics. Research on the Pitman–Yor process was initiated by Pitman and Yor (1992), where it was used to study the ranked lengths of excursions of a Markov process. Later on, Perman et al. (1992) demonstrated the connection between the random weights of a Pitman–Yor process and the normalized jumps of a stable process. Pitman (1995) introduced the partially exchangeable probability function and derived a sampling formula for the Pitman–Yor process. Pitman (2006) designed a generative approach for sampling from the Pitman–Yor process, which is known as the two-parameter Chinese restaurant process. The application of the Pitman–Yor process has been found in many areas. For example, Favaro et al. (2009) derived a species sampling formula from the Pitman–Yor process and constructed a Bayesian nonparametric methodology to deal with the prediction issue within the species sampling problems; Jara et al. (2010) introduced a linear-dependent model based on the Pitman–Yor process and used it to analyze a dataset generated by a dental longitudinal study; Carmona et al. (2019) developed a Bayesian nonparametric mixture model based on the Pitman–Yor process prior and studied its posterior inference method. Other applications can be found in bioinformatics and computational biology (Lijoi et al., 2007, 2008), image segmentation (Sudderth & Jordan, 2008), curve estimation (Jara et al., 2010), gene networks inference (Ni et al., 2018), and econometrics (Bassetti et al., 2014).

The Pitman–Yor process is a random probability measure of the form

$$\tilde{\mathcal{P}}_{\infty}^{\alpha,\theta}(.) = \sum_{i=1}^{\infty} \tilde{p}_i \delta_{K_i}(.), \tag{1}$$

where $\delta_{K_i}(.)$ is a point mass at K_i, $\{K_i\}_{i\geq 1}$ are independent and identically distributed random variables with the distribution H on a Polish space $\mathcal{X}$, and $\{\tilde{p}_i\}_{i\geq 1}$ are random weights such that $\tilde{p}_i \geq 0$ and $\sum_{i\geq 1} \tilde{p}_i = 1$, independent of $\{K_i\}_{i\geq 1}$. For simplicity, we will omit the hyperparameters α and θ where possible. The random weights admit the following stick-breaking representation,

$$\tilde{p}_i \overset{d}{=} V_i \prod_{j=1}^{i-1}(1-V_j) \quad \text{and} \quad V_j \sim \text{beta}(1-\alpha, \theta + j\alpha), \quad i = 1, 2, 3, \ldots, \tag{2}$$

with the discount parameter $0 \leq \alpha < 1$ and the concentration parameter $\theta > -\alpha$. By setting $\alpha = 0$, we revert to the stick-breaking representation of a Dirichlet process. Throughout this paper, we use the tilde notation to emphasize that the sequence $\{\tilde{p}_i\}_{i\geq 1}$ is presented in its original order. In other words, it follows Equation (2) without any reordering. Based on the stick-breaking representation, several approximation methods have been proposed for the Pitman–Yor process. Ishwaran and James (2001) introduced the truncated stick-breaking process with a fixed truncation level N,

$$\tilde{\mathcal{P}}_N(.) = \sum_{i=1}^{N} \tilde{p}_i \delta_{K_i}(.) + \tilde{e}_N \delta_{K_0}(.), \tag{3}$$

where $\tilde{p}_1, \ldots, \tilde{p}_N$ are defined as (2), $\tilde{e}_N := 1 - \tilde{p}_1 - \cdots - \tilde{p}_N$ such that $\tilde{\mathcal{P}}_N$ is a valid random probability measure, and K_0 has the distribution H independently. As $N \to \infty$, $\tilde{\mathcal{P}}_N$ converges to $\tilde{\mathcal{P}}_{\infty}$ in total variation distance almost surely (see, e,g“ Sec. 4.3.3 of Ghosal & van der Vaart, 2017),

hence a finite approximation for the distribution of the Pitman–Yor process is obtained. From the definition, it is clear that the approximation error of this method is characterized by the tail probability $\tilde{e}_N$. A study about the expectation of the tail probability can be found in Ishwaran and James (2001). On the other hand, Arbel et al. (2019) introduced a random stopping rule which can achieve an almost sure control on the approximation error in total variation distance. Specifically, they used the random truncation level

$$N(\epsilon) := \min\{n \geq 1 | \tilde{e}_n < \epsilon\}.$$

This leads to the ϵ-Pitman–Yor process

$$\tilde{\mathcal{P}}_\epsilon(.) = \sum_{i=1}^{N(\epsilon)} \tilde{p}_i \delta_{K_i}(.) + \tilde{e}_{N(\epsilon)} \delta_{K_0}(.).$$

The ϵ-Pitman–Yor process generalizes the ϵ-Dirichlet process proposed by Muliere and Tardella (1998). It follows from the definition that the random stopping rule controls the approximation error according to the total variation bound

$$d_{TV}(\tilde{\mathcal{P}}_\infty, \tilde{\mathcal{P}}_\epsilon) = \sup_{A \subset \mathcal{X}} |\tilde{\mathcal{P}}_\infty(A) - \tilde{\mathcal{P}}_\epsilon(A)| \leq \epsilon,$$

almost surely (see Prop. 4.20 of Ghosal & van der Vaart, 2017). The construction also guarantees the almost sure convergence of the measurable functionals of $\tilde{\mathcal{P}}_\epsilon$ to that of $\tilde{\mathcal{P}}_\infty$ as $\epsilon \to 0$. Numerical implementation of the functional estimation can be found in Arbel et al. (2019). Finally, Al Labadi and Zarepour (2014) proposed a different approximation method for the Pitman–Yor process based on the decreasing sequences of random weights of a gamma process and a stable process. They also provided a simulation scheme for the approximation.

In this paper, we propose a new random probability measure called the truncated two-parameter Poisson–Dirichlet process and use it to approximate the distribution of the Pitman–Yor process. Our construction is based on the ranked random weights of the Pitman–Yor process. Since $\{K_i\}_{i\geq 1}$ are i.i.d., it is clear that the Pitman–Yor process is invariant in distribution to a reordering of $\{\tilde{p}_i\}_{i\geq 1}$. In particular, let $p_1 > p_2 > \ldots$ be the ranked values of $\{\tilde{p}_i\}_{i\geq 1}$, then the random probability measure

$$\mathcal{P}_\infty(.) = \sum_{i=1}^{\infty} p_i \delta_{K_i}(.), \tag{4}$$

is identical in distribution to the Pitman–Yor process (1). We call (4) a two-parameter Poisson–Dirichlet process as the ranked random weights $\{p_i\}_{i\geq 1}$ follow the two-parameter Poisson–Dirichlet distribution (see Pitman & Yor, 1997). In fact, the original sequence $\{\tilde{p}_i\}_{i\geq 1}$ is a size-biased permutation of $\{p_i\}_{i\geq 1}$, see McCloskey (1965) for more details. To derive a finite approximation to the Pitman–Yor process, we truncate the two-parameter Poisson–Dirichlet process at the fixed truncation level N. Then we get the truncated two-parameter Poisson–Dirichlet process

$$\mathcal{P}_N(.) = \sum_{i=1}^{N} p_i \delta_{K_i}(.) + e_N \delta_{K_0}(.),$$

where $e_N := \sum_{i=N+1}^{\infty} p_i$. It is easy to see that $\mathcal{P}_N$ has a lower approximation error compared to $\tilde{\mathcal{P}}_N$ because the tail probability e_N is almost surely smaller than $\tilde{e}_N$ for the same truncation level N. It follows that $\sup_{A\subset\mathcal{X}} |\tilde{\mathcal{P}}_\infty(A) - \mathcal{P}_N(A)| \leq \sup_{A\subset\mathcal{X}} |\tilde{\mathcal{P}}_\infty(A) - \tilde{\mathcal{P}}_N(A)|$ almost surely. Thus, we expect the truncated two-parameter Poisson–Dirichlet process to provide a better approximation to the Pitman–Yor process and its functionals. This will be examined by a comparison study based on the truncation error and estimation accuracy.

Our approximation method is similar to the Ferguson–Klass representation of a random probability measure. Recall that a completely random measure (CRM) induced by the rate measure $\nu(dw)$ can be written as

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

where $J_i := \nu^{\leftarrow}(\Gamma_i)$, $\nu^{\leftarrow}(w) := \inf\{x|\nu([x,\infty)) \leq w\}$, $\Gamma_i := \sum_{j=1}^{i} E_j$, and $E_j \sim Exp(1)$ are i.i.d. exponential random variables. We can then derive a random probability measure from the CRM by normalization, that is, by considering $\Theta/\Theta(\mathcal{X})$. Examples of random probability measures derived from the normalized CRM include the Dirichlet process, the normalized inverse-Gaussian process and the generalized gamma process. Since the sequence $\{J_i\}_{i\geq 1}$ is decreasing, the Ferguson–Klass representation is based on a decreasing sequence of random weights. This means the truncated random probability measure $\Theta_N/\Theta(\mathcal{X})$, where

$$\Theta_N(.) := \sum_{i=1}^{N} J_i \delta_{K_i}(.) + \theta_N \delta_{K_0}(.) \quad \text{and} \quad \theta_N := \sum_{i=N+1}^{\infty} J_i,$$

has the lowest truncation error compared to other approximations. See Campbell et al. (2019) for further discussion. Although the Pitman–Yor process is not a normalized CRM, its connection to the stable process CRM has been revealed by a change of measure in Pitman and Yor (1997). Thus, we will construct the truncated Ferguson–Klass representation of a stable process and derive the truncated two-parameter Poisson–Dirichlet process from it via the change of measure.

A typical application of the finite approximation of the Pitman–Yor process is the posterior inference of the Pitman–Yor process mixture models (PYMM). Consider the mixture model

$$\begin{aligned} (X_i|Y_i) &\sim \mathcal{K}(X_i|Y_i), \quad i = 1,\ldots,n, \\ (Y_i|\tilde{P}) &\overset{iid}{\sim} \tilde{P}, \\ \tilde{P} &\sim \tilde{\mathcal{P}}_\infty, \end{aligned} \tag{5}$$

where $X_1,\ldots,X_n$ represent the observations, they are assumed to be independent conditional on the latent variables Y_i, $\mathcal{K}(X_i|Y_i)$ denotes the conditional distribution of X_i given Y_i, the latent variables Y_i are i.i.d. with the distribution $\tilde{P}$, and $\tilde{P}$ is a sample of the Pitman–Yor process prior. The joint distribution of $\mathbf{Y} = (Y_1,\ldots,Y_n)$ is characterized by the predictive distribution of the Pitman–Yor process

$$\mathbb{P}(Y_{i+1}|Y_1,\ldots,Y_i) = \frac{\theta + m_i\alpha}{\theta + i}\delta_H + \sum_{j=1}^{m_i} \frac{n_j - \alpha}{\theta + i}\delta_{K_j^*}, \quad i = 1,2,3,\ldots, \tag{6}$$

where m_i is the number of distinct values K_j^* observed in the first i draws, and n_j is the number of latent variables taking the value of K_j^* such that $\sum_{j=1}^{m_i} n_j = i$. The predictive distribution implies a Pólya urn scheme for sampling from the posterior of the Pitman–Yor process, which leads to the marginalization method for the posterior inference of PYMM. See Ishwaran and James (2001) for the details. The marginalization method is easy to use, but it also suffers from the side effects of slow mixing of the Markov chain and allowing the inference to be based only on the latent variables. To avoid these limitations, Ishwaran and James (2001) replaced the Pitman–Yor process prior $\tilde{\mathcal{P}}_\infty$ in model (5) by a truncated stick-breaking process $\tilde{\mathcal{P}}_N$ and developed a blocked Gibbs sampler for posterior inference. They showed that the truncation method provides a good approximation to the PYMM. In this paper, we will provide a new approximation to the PYMM using the truncated two-parameter Poisson–Dirichlet process. Based on a lower-error approximation, our approach is expected to have better inferential quality. We will verify this by a comparison study.

The rest of the paper is organized as follows. Section 2 constructs the truncated two-parameter Poisson–Dirichlet process and studies its truncation error. Section 3 designs the simulation algorithms for the truncated two-parameter Poisson–Dirichlet process and uses the algorithms to estimate the functionals of the Pitman–Yor process. Section 4 adapts the truncated two-parameter Poisson–Dirichlet process into the Pitman–Yor process mixture model and develops a posterior inference scheme. Section 5 presents some numerical results. Section 6 summarizes the paper with a discussion of open questions. The supplementary materials contain the derivations of the main results.

2 | CONSTRUCTION AND DISTRIBUTIONAL PROPERTIES

In this section, we construct the truncated two-parameter Poisson–Dirichlet process and study its distributional properties. For a positive integer N, a truncated two-parameter Poisson–Dirichlet process is a random probability measure defined as

$$\mathcal{P}_N^{\alpha,\theta}(.) = \sum_{i=1}^{N} p_i \delta_{K_i}(.) + e_N \delta_{K_0}(.), \tag{7}$$

where $0 < \alpha < 1, \theta > -\alpha$, $\{p_i\}_{i\geq 1}$ denotes the components of the two-parameter Poisson–Dirichlet distribution $\mathrm{PD}(\alpha, \theta)$, $e_N = \sum_{i=N+1}^{\infty} p_i$, $\delta_{K_i}(.)$ is a point mass at K_i, and $\{K_i\}_{i\geq 0}$ are i.i.d. random variables with the distribution H. From the definition, it is clear that the construction of the truncated two-parameter Poisson–Dirichlet process hinges on the components of the two-parameter Poisson–Dirichlet distribution. To obtain these components, recall that Pitman and Yor (1997) gave a connection between the $\mathrm{PD}(\alpha, 0)$ distribution and the stable process. Consider a stable process τ_s with the Lévy measure $\nu(dw) = \alpha w^{-\alpha-1} \mathbb{1}_{\{0<w<\infty\}} dw$. The Lévy–Khintchine representation of τ_s is

$$\mathbb{E}(\exp(-\beta\tau_s)) = \exp\left(-s\int_0^\infty (1 - e^{-\beta w})\alpha w^{-\alpha-1} dw\right).$$

Let $\{J_i\}_{i\geq 1}$ be the ranked jumps of τ_s on the time interval $[0, t]$, such that $J_1 > J_2 > \dots$ and $\tau_t = \sum_{i=1}^{\infty} J_i$, then the normalized jumps $(J_1/\tau_t, J_2/\tau_t, \dots)$ follow the $\mathrm{PD}(\alpha, 0)$ distribution.

Thus, we get a special case of the two-parameter Poisson–Dirichlet process in terms of $\mathcal{P}_{\infty}^{\alpha,0}(.) = \sum_{i=1}^{\infty}(J_i/\tau_t)\delta_{K_i}(.)$, and a truncation can be made based on this representation. To generalize this method to any $\theta > -\alpha$, recall that Pitman and Yor (1997) provided a change of measure between the components of the PD$(\alpha, 0)$ and PD(α, θ) distributions. Next, we use this method to construct the truncated two-parameter Poisson–Dirichlet process.

To facilitate the construction, we first prepare some preliminary results about the jumps of a stable process. Note that to construct a random probability measure, we do not need the whole time history of the stable process, so any compact interval will suffice, and we pick $[0, 1]$. Let $\{J_i\}_{i\geq 1}$ be the ranked jumps of a stable process on the time interval $[0, 1]$. Denote by ${}^{(N)}\tau_1$ the N-trimmed subordinator (Ipsen & Maller, 2017) derived from τ_1, that is, the process obtained by removing the N largest jumps from τ_1, such that ${}^{(N)}\tau_1 := \tau_1 - \sum_{i=1}^{N} J_i$. The following lemma derives the conditional density of ${}^{(N)}\tau_1$.

Lemma 1 (Conditional density of N-trimmed stable process). *Conditioning on the N-th largest jump J_N of the stable process τ_1, the density of the N-trimmed stable process ${}^{(N)}\tau_1$ is given by*

$$f_{\alpha,J_N}(w)dw = J_N^{-1} g_Z\left(wJ_N^{-1}; \alpha, J_N^{-\alpha}\right)dw,$$

where $g_Z(z; \alpha, t)$ is defined as

$$g_Z(z; \alpha, t) = \sum_{i=0}^{n-1} \frac{(-t)^i}{i!} L_i(z; \alpha, t),$$

for $n - 1 < z \leq n$, $n = 1, 2, \ldots$, and $L_i(z; \alpha, t)$ is defined recursively as follows,

$$L_0(z; \alpha, t) = e^t \frac{1}{\pi} \int_0^{\pi} \frac{\alpha}{1-\alpha} A(u) z^{-\frac{1}{1-\alpha}} (t\Gamma(1-\alpha))^{\frac{1}{1-\alpha}} e^{-A(u)z^{-\frac{\alpha}{1-\alpha}}(t\Gamma(1-\alpha))^{\frac{1}{1-\alpha}}} du, \quad z > 0,$$

and

$$L_{i+1}(z; \alpha, t) = \int_1^{z-i} L_i(z - s; \alpha, t)\alpha s^{-\alpha-1} ds, \quad z > i + 1,$$

where

$$A(u) := \left[\frac{\sin(\alpha u)^{\alpha} \sin((1-\alpha)u)^{1-\alpha}}{\sin(u)}\right]^{\frac{1}{1-\alpha}}.$$

To illustrate the result of Lemma 1, we plot the density of ${}^{(N)}\tau_1$ 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 ${}^{(N)}\tau_1$ numerically at some fixed points and plot the results as stars. From the figures we can see that both methods produce similar results. Next, we use this lemma to derive the joint density of the N largest jumps and the sum of the smaller jumps of a stable process.

Lemma 2. *Let $J_1 > J_2 > \ldots$ be the ranked jumps of a stable process on the time interval $[0, 1]$. Denote by $R_k := J_{k+1}/J_k$ the ratio between the $(k + 1)$th and kth largest jumps,*

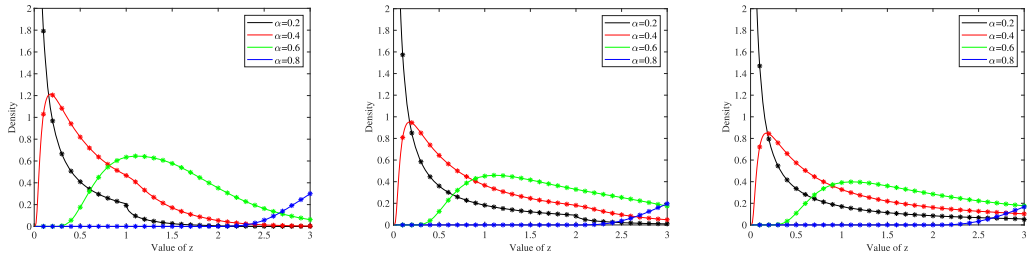

FIGURE 1 Density plot for the N-trimmed stable process ${}^{(N)}\tau_1$ using Lemma 1 (in solid line) and the numerical inverse Laplace transform results using the CME method (in stars). The parameter values are $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$ and $J_N \in \{1, 2, 3\}$. (a) $J_N = 1$; (b) $J_N = 2$; (c) $J_N = 3$.

then the joint density of $\left(J_1, R_1, \ldots, R_{N-1}, {}^{(N)}\tau_1\right)$ *is*

$$\begin{aligned}&\mathbb{P}\left(J_1 \in dx_1, R_1 \in dr_1, \ldots, R_{N-1} \in dr_{N-1}, {}^{(N)}\tau_1 \in dy\right)\\&\quad = \exp\left(-(x_1 r_1 \ldots r_{N-1})^{-\alpha}\right) \alpha^N x_1^{-N\alpha-1} r_1^{-(N-1)\alpha-1} \ldots r_{N-1}^{-\alpha-1} f_{\alpha, x_1 r_1 \ldots r_{N-1}}(y) dy dx_1 dr_1 \ldots dr_{N-1},\end{aligned}$$

where $x_1 \in (0, \infty)$, $r_k \in (0, 1)$, $k = 1, \ldots, N-1$, $y \in (0, \infty)$, *and* $f_{\alpha, x_1 r_1 \ldots r_{N-1}}(y)$ *denotes the density of the N-trimmed stable process* ${}^{(N)}\tau_1$.

Using Lemma 2, we can construct the truncated Ferguson–Klass representation of the CRM induced by the stable process, and hence the truncated two-parameter Poisson–Dirichlet process with $\theta = 0$. The following theorem generalizes this method to an arbitrary $\theta > -\alpha$. It also provides a representation in law for the random weights of the truncated two-parameter Poisson–Dirichlet process, which can be used in the numerical implementation.

Theorem 1. *The N largest components of a two-parameter Poisson–Dirichlet distribution have the representation*

$$p_i = \frac{J_1 \prod_{j=1}^{i-1} R_j}{J_1 + J_1 R_1 + \cdots + J_1 R_1 \ldots R_{N-1} + {}^{(N)}\tau_1}, \tag{8}$$

for $i = 1, \ldots, N$, *and the sum of the smaller components can be represented by*

$$e_N = \sum_{i=N+1}^{\infty} p_i = \frac{{}^{(N)}\tau_1}{J_1 + J_1 R_1 + \cdots + J_1 R_1 \ldots R_{N-1} + {}^{(N)}\tau_1}. \tag{9}$$

The random variables $(J_1, R_1, \ldots, R_{N-1}, {}^{(N)}\tau_1)$ *have the joint density*

$$\begin{aligned}&\mathbb{P}(J_1 \in dx_1, R_1 \in dr_1, \ldots, R_{N-1} \in dr_{N-1}, {}^{(N)}\tau_1 \in dy)\\&= \frac{\Gamma(\theta+1)\Gamma(1-\alpha)^{\theta/\alpha}}{\Gamma(\theta/\alpha+1)} \frac{\exp(-(x_1 r_1 \ldots r_{N-1})^{-\alpha}) \alpha^N x_1^{-N\alpha-1} r_1^{-(N-1)\alpha-1} \ldots r_{N-1}^{-\alpha-1}}{(x_1 + x_1 r_1 + \cdots + x_1 r_1 \ldots r_{N-1} + y)^{\theta}}\\&\quad \times f_{\alpha, x_1 r_1 \ldots r_{N-1}}(y) dy dx_1 dr_1 \ldots dr_{N-1},\end{aligned} \tag{10}$$

where $x_1 \in (0, \infty)$, $r_k \in (0, 1)$, $k = 1, \ldots, N-1$, $y \in (0, \infty)$, *and* $f_{\alpha, x_1 r_1 \ldots r_{N-1}}(y)$ *denotes the density of the N-trimmed stable process* ${}^{(N)}\tau_1$.

Next, we investigate the truncation error e_N of the truncated two-parameter Poisson–Dirichlet process. Since $\{p_i\}_{i\geq 1}$ are the ranked values of the stick-breaking random weights $\{\tilde{p}_i\}_{i\geq 1}$, it is clear that $\sum_{i=1}^{N} p_i \geq \sum_{i=1}^{N} \tilde{p}_i$ a.s. Combining this with the fact that $\sum_{i=1}^{\infty} p_i = \sum_{i=1}^{\infty} \tilde{p}_i = 1$, we have $e_N \leq \tilde{e}_N$ a.s. Thus, the truncated two-parameter Poisson–Dirichlet process has a lower truncation error compared to the truncated stick-breaking process almost surely. To further investigate the truncation error, we use the representation (9) to express e_N in terms of

$$\begin{aligned}&\{e_N = y | J_1, R_1, \ldots, R_{N-1}\}\\ &\quad = \left\{ {}^{(N)}\tau_1 = \frac{y}{1-y}(x_1 + x_1 r_1 + \cdots + x_1 r_1 \ldots r_{N-1}) | J_1 = x_1, R_1 = r_1, \ldots, R_{N-1} = r_{N-1} \right\},\end{aligned}$$

where $y \in (0,1)$. Using the joint density of $J_1, R_1, \ldots, R_{N-1}$, it is possible to integrate out the condition to derive the explicit distribution of e_N. However, the integral is too complicated to calculate explicitly. Handa (2009) made a significant contribution to this problem by deriving the joint density of the random weights $(p_1, \ldots, p_N)$, and the probability $\mathbb{P}(e_N > y)$ could be obtained by integrating the joint density over $\nabla_N(y) := \{p_1 > 0, \cdots, p_N > 0, p_1 + \cdots + p_N \leq y\}$, while the explicit expression of the probability remains an open question. For this reason, we focus on the expectation of the truncation error. From Prop. 17 of Pitman and Yor (1997), we know

$$\mathbb{E}(p_n) = \frac{\Gamma(1-\alpha)^{\theta/\alpha}\Gamma(\theta/\alpha + n)}{\Gamma(n)\Gamma(\theta/\alpha + 1)} \int_0^{\infty} t^{\theta} e^{-t} \phi_{\alpha}(t)^{n-1} \psi_{\alpha}(t)^{-\theta/\alpha - n} dt, \tag{11}$$

where

$$\phi_{\alpha}(\lambda) = \alpha \int_1^{\infty} e^{-\lambda x} x^{-\alpha - 1} dx \quad \text{and} \quad \psi_{\alpha}(\lambda) = \Gamma(1-\alpha)\lambda^{\alpha} + \phi_{\alpha}(\lambda).$$

Then the expectation of the tail probability can be expressed as $\mathbb{E}(e_N) = 1 - \sum_{i=1}^{N} \mathbb{E}(p_i)$. On the other hand, since

$$\mathbb{E}(\tilde{p}_n) = \left(1 - \frac{1-\alpha}{(1-\alpha) + (\theta + \alpha)}\right) \ldots \left(1 - \frac{1-\alpha}{(1-\alpha) + (\theta + (n-1)\alpha)}\right) \frac{1-\alpha}{(1-\alpha) + (\theta + n\alpha)}, \tag{12}$$

the expectation of the tail probability of the stick-breaking process is given by $\mathbb{E}(\tilde{e}_N) = 1 - \sum_{i=1}^{N} \mathbb{E}(\tilde{p}_i)$. We plot the difference between the expectations of the tail probabilities, that is, $\mathbb{E}(\tilde{e}_N) - \mathbb{E}(e_N)$, in Figure 2. The figures illustrate the improvement of the prior approximation quality by using the ranked random weights. We can see that the improvement is more significant for small N and large θ. On the other hand, we plot the mean truncation error with different parameters in Figure 3. The figures show that the truncated stick-breaking process requires more terms to achieve the same prior approximation accuracy as the truncated two-parameter Poisson–Dirichlet process.

3 | PRIOR SIMULATION ALGORITHMS

In this section, we design two simulation algorithms for the truncated two-parameter Poisson–Dirichlet process and use them to estimate the functionals of the Pitman–Yor process

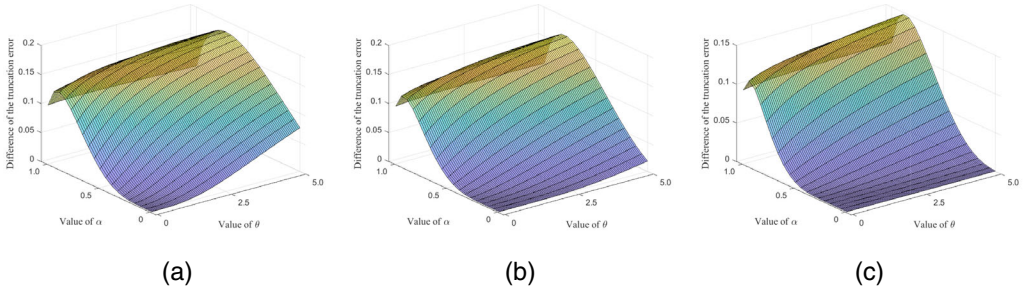

FIGURE 2 Difference between the mean truncation error of the stick-breaking process and the two-parameter Poisson–Dirichlet process. The results are obtained by evaluating $\mathbb{E}(\tilde{e}_N) - \mathbb{E}(e_N)$ numerically using Equations (11) and (12). The parameter values are $\alpha \in (0,1)$, $\theta \in (0,5)$ and $N \in \{10, 20, 50\}$. (a) $N = 10$; (b) $N = 20$; (c) $N = 50$.

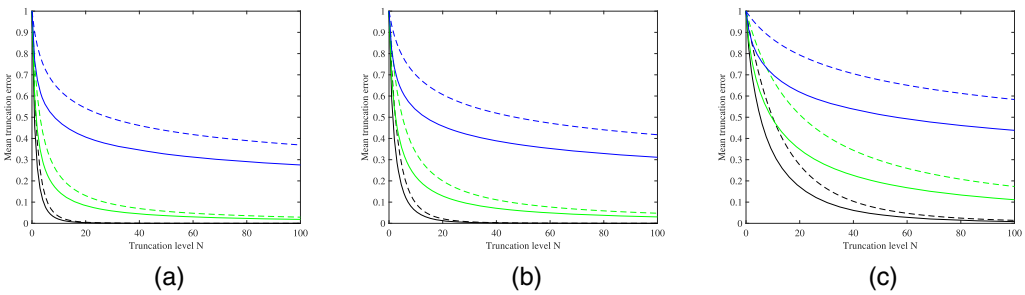

FIGURE 3 Comparison between the mean truncation error of the stick-breaking process (in dashed curves) and the two-parameter Poisson–Dirichlet process (in solid curves) using Equations (11) and (12). The value of α is denoted by the color of the curves, $\alpha = 0.2$ (black), 0.5 (green) and 0.8 (blue). (a) $\theta = 1$; (b) $\theta = 2$; (c) $\theta = 10$.

numerically. It follows from Theorem 1 that to sample from the random weights of the truncated two-parameter Poisson–Dirichlet process, we only need to simulate the random vector $(J_1, R_1, \ldots, R_{N-1}, {}^{(N)}\tau_1)$. Next, we develop an algorithm for this purpose. The algorithm involves the simulation of the N-trimmed stable process ${}^{(N)}\tau_1$, which can be done via Alg. 4.3 of Dassios et al. (2020). We denote this algorithm by AlgorithmTS(.,.) and attach the full steps in the Appendix S1.

Algorithm 1. *When $\theta > 0$, the simulation algorithm for the random vector $(J_1, R_1, \ldots, R_{N-1}, {}^{(N)}\tau_1)$ is given as follows.*

1. *For every $i = 1, \ldots, N-1$, sample from a beta distribution $R_i \leftarrow Beta(\theta + i\alpha, 1)$.*
2. *Sample from a gamma distribution $Z \leftarrow Gamma(\theta/\alpha + N, 1)$.*
3. *Sample from a N-trimmed stable process ${}^{(N)}\tau_1 \leftarrow Z^{-1/\alpha} \times AlgorithmTS(\alpha, \Gamma(1-\alpha)Z)$.*
4. *Sample from a uniform distribution $U \leftarrow U[0,1]$, if*

$$U \leq \left[(1 + R_1 + \cdots + R_1 \ldots R_{N-1}) + Z^{1/\alpha}(R_1 \ldots R_{N-1})^{(N)}\tau_1\right]^{-\theta},$$

accept the candidates and output $J_1 \leftarrow Z^{-1/\alpha}(R_1 \ldots R_{N-1})^{-1}, R_1, \ldots, R_{N-1}, {}^{(N)}\tau_1$. Otherwise, return to Step 1.

Algorithm 1 uses the acceptance-rejection method, so we need to investigate its acceptance rate. From the derivation of Algorithm 1 (see Appendix S1) we know the expected number of iterations for an acceptance is $M = M(\alpha, \theta) := \Gamma(\theta + 1)\Gamma(1 - \alpha)^{\theta/\alpha}$. It follows that the efficiency of the algorithm suffers less from the truncation level N, although a large N requires more beta random numbers, but more from α and θ. Since $M(\alpha, \theta)$ is increasing in both variables, Algorithm 1 works better with small α and θ. When these parameters are large, the algorithm becomes computationally expensive. To avoid this pitfall, we provide an alternative algorithm using the MCMC method. This algorithm is simply described as running Algorithm 4, which will be explained later, iteratively with an empty set of observations ($n = 0$). The idea is to draw from the joint density of $(J_1, R_1, \ldots, R_{N-1}, {}^{(N)}\tau_1)$ using the MCMC method. This approach suffers less from the values of the parameters. Next, we use Algorithm 1 and Algorithm 4 to sample from the five largest components of the two-parameter Poisson–Dirichlet distribution. To achieve a fair comparison, we run both algorithms for a fixed time duration of 30 s. When $\theta = 10$, the exact simulation algorithm is too slow, and we only use the MCMC algorithm. The sample averages are recorded in Table 1. The table shows that both algorithms can achieve a reasonable level of Monte Carlo error. We remark that the MCMC algorithm completes the study in Dassios and Zhang (2021), where the large values of θ were considered only when θ/α is an integer.

The truncated two-parameter Poisson–Dirichlet process and its simulation algorithms can be used to estimate the functionals of the Pitman–Yor process. We demonstrate this method with the cumulative distribution function F of the Pitman–Yor process. Denote by $\tilde{F}_N$ and F_N the cdf of $\tilde{\mathcal{P}}_N$ and $\mathcal{P}_N$. From the total variation bound, we know $|F(.) - \tilde{F}_N(.)| < \tilde{e}_N$ and $|F(.) - F_N(.)| < e_N$. Thus, we can estimate the cdf of the Pitman–Yor process by that of the truncated stick-breaking

TABLE 1 Monte Carlo estimation for $\mathbb{E}(p_i)$, $i = 1, \ldots, N$, and $\mathbb{E}(e_N)$ using Algorithm 1 and Algorithm 4.

	Algorithm	p_1	p_2	p_3	p_4	p_5	e_N	Sample size
$\alpha = 0.2, \theta = 1.0$	True mean	0.5408	0.1970	0.0970	0.0545	0.0332	0.0774	N.A.
	Algorithm 1	0.5406	0.1974	0.0971	0.0545	0.0332	0.0772	219,532
	Algorithm 4 ($n = 0$)	0.5399	0.1979	0.0975	0.0546	0.0331	0.0770	41,547
$\alpha = 0.5, \theta = 1.0$	True mean	0.4028	0.1574	0.0881	0.0573	0.0406	0.2537	N.A.
	Algorithm 1	0.4034	0.1579	0.0885	0.0574	0.0407	0.2522	203,490
	Algorithm 4 ($n = 0$)	0.4043	0.1587	0.0885	0.0572	0.0406	0.2507	49,603
$\alpha = 0.8, \theta = 1.0$	True mean	0.2322	0.0898	0.0530	0.0368	0.0278	0.5604	N.A.
	Algorithm 1	0.2317	0.0897	0.0530	0.0368	0.0278	0.5610	76,953
	Algorithm 4 ($n = 0$)	0.2365	0.0903	0.0532	0.0367	0.0275	0.5559	40,965
$\alpha = 0.2, \theta = 10.0$	True mean	0.1726	0.1097	0.0823	0.0659	0.0547	0.5148	N.A.
	Algorithm 4 ($n = 0$)	0.1717	0.1096	0.0825	0.0657	0.0556	0.5149	17,616
$\alpha = 0.5, \theta = 10.0$	True mean	0.1353	0.0834	0.0619	0.0495	0.0412	0.6286	N.A.
	Algorithm 4 ($n = 0$)	0.1340	0.0832	0.0614	0.0490	0.0404	0.6321	30,057
$\alpha = 0.8, \theta = 10.0$	True mean	0.0866	0.0493	0.0354	0.0278	0.0229	0.7781	N.A.
	Algorithm 4 ($n = 0$)	0.0853	0.0477	0.0339	0.0262	0.0213	0.7856	23,467

Notes: The true means are derived from the numerical evaluation of Equation (11). The parameter values are $\alpha \in \{0.2, 0.5, 0.8\}$, $\theta \in \{1, 10\}$, $N = 5$, and the running time is 30 s for all the simulation algorithms and parameter settings.

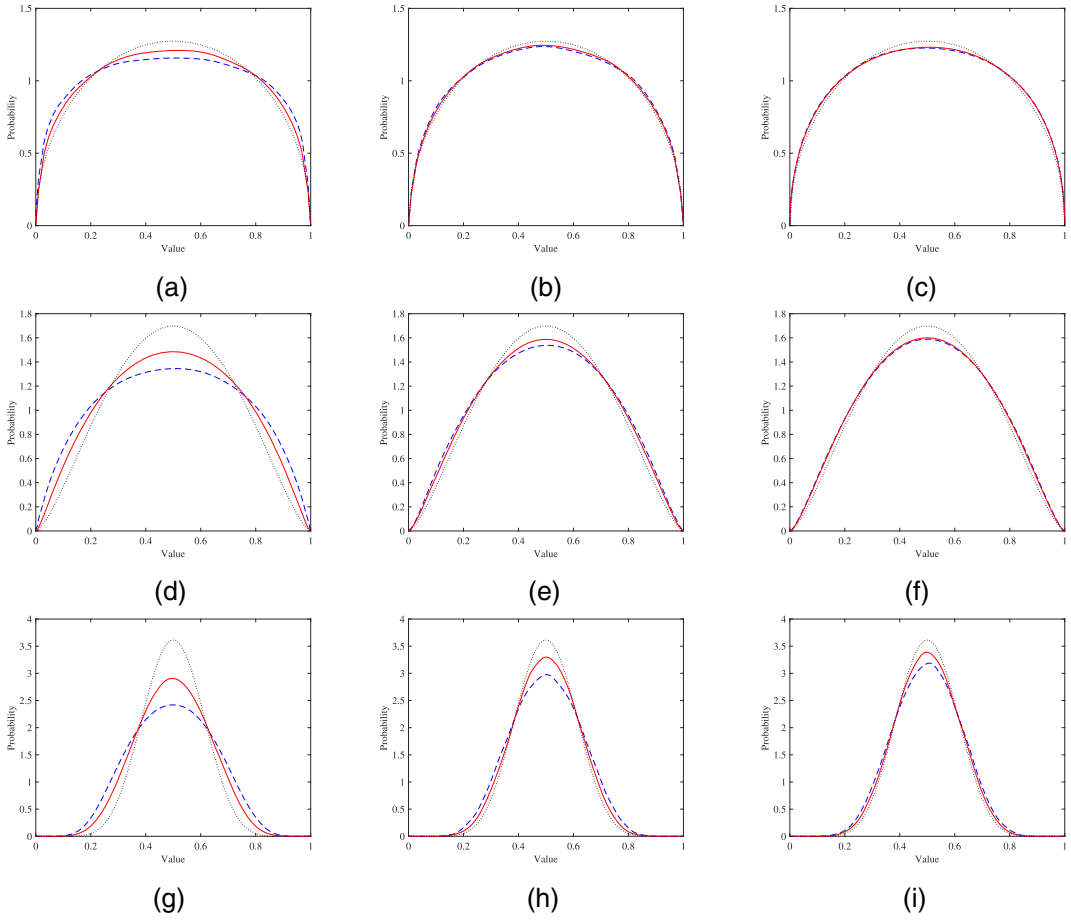

FIGURE 4 Density plot for the random probability $F(1/2)$ using Algorithm 1 (in red solid line, $\theta = 1, 2$), Algorithm 4 (in red solid line, $\theta = 10$) and the truncated stick-breaking process (in blue dashed curve, using Equation 3) to approximate the Pitman–Yor process. The sample size is 10^4. The density under the Pitman–Yor process is the black dotted line. The parameter α is fixed at 0.5, θ is, respectively, equal to $\{1, 2, 10\}$ on the first, second and third row, and $N \in \{20, 50, 100, 150, 200\}$. (a) $\alpha = 0.5, \theta = 1, N = 20$; (b) $\alpha = 0.5, \theta = 1, N = 50$; (c) $\alpha = 0.5, \theta = 1, N = 100$; (d) $\alpha = 0.5, \theta = 2, N = 20$; (e) $\alpha = 0.5, \theta = 2, N = 50$; (f) $\alpha = 0.5, \theta = 2, N = 100$; (g) $\alpha = 0.5, \theta = 10, N = 100$; (h) $\alpha = 0.5, \theta = 10, N = 150$; (i) $\alpha = 0.5, \theta = 10, N = 200$.

process and the truncated two-parameter Poisson–Dirichlet process. On the other hand, the exact distribution of F has been derived in James et al. (2010). In particular, if $\alpha = 0.5$ and H is a uniform distribution on $[0, 1]$, then $F(1/2) \sim \text{Beta}(\theta + 1/2, \theta + 1/2)$, and the density of $F(1/3)$ is given by

$$f(w)dw = \frac{2}{\sqrt{\pi}} 9^{\theta} \frac{\Gamma(\theta + 1)}{\Gamma(\theta + 1/2)} \frac{(w(1-w))^{\theta - 1/2}}{(1 + 3w)^{\theta + 1}} dw.$$

In Figure 4, we use Algorithm 1 and Algorithm 4 to draw from $F_N(1/2)$ so to illustrate that the estimation becomes more accurate as the truncation level grows higher. As for $F(1/3)$, we record the Kolmogorov distance between the exact and estimated distributions, together with their quantiles, in Table 2. The results suggest a lower Kolmogorov distance arising from the truncated two-parameter Poisson–Dirichlet process compared to the truncated stick-breaking process.

TABLE 2 Summary statistics for $F(1/3)$ using Algorithm 1 (PD, $\theta = 1, 2$), Algorithm 4 (PD, $\theta = 10, 20$) and the truncated stick-breaking process (SB, using Equation 3) to approximate the Pitman–Yor process.

		d_K		25%			Median			75%		
θ	N	SB	PD	SB	PD	PY	SB	PD	PY	SB	PD	PY
1	50	7.69	5.12	0.1365	0.1369	0.1394	0.2816	0.2817	0.2821	0.4917	0.4912	0.4890
1	100	3.94	3.68	0.1392	0.1393	0.1394	0.2833	0.2823	0.2821	0.4918	0.4907	0.4890
2	50	18.74	15.04	0.1748	0.1825	0.1813	0.3059	0.3025	0.3017	0.4657	0.4642	0.4576
2	100	11.43	10.95	0.1785	0.1800	0.1813	0.3014	0.3014	0.3017	0.4591	0.4493	0.4576
10	100	93.30	39.85	0.2315	0.2503	0.2603	0.3192	0.3257	0.3255	0.4236	0.4126	0.3981
10	200	34.28	23.21	0.2517	0.2560	0.2603	0.3255	0.3240	0.3255	0.4074	0.3991	0.3981
20	100	282.89	150.73	0.2063	0.2447	0.2817	0.2822	0.3194	0.3293	0.4726	0.4192	0.3807
20	200	136.34	82.30	0.2479	0.2655	0.2817	0.3185	0.3236	0.3293	0.4140	0.3948	0.3807

Notes: The Kolmogorov distance (d_K) is between the cumulative distribution function of $F(1/3)$ and the empirical cdf (multiplied by a factor of 1000). The parameter values are $\alpha = 0.5$, $\theta \in \{1, 2, 10, 20\}$, $N \in \{50, 100, 200\}$, and the running time is 30 s for all the simulation algorithms and parameter settings.

Recall that the same experiments were carried out by Arbel et al. (2019) for the ϵ-Pitman–Yor process, and we can compare our numerical results to Fig. 2 and Table 3 of Arbel et al. (2019). For example, when $\alpha = 0.5$, $\theta = 1$ and $N = 50$, the Kolmogorov distance between the exact distribution of $F(1/3)$ and the estimation $F_N(1/3)$ is 0.00512, while the distance between the exact distribution and the estimated curve derived from the ϵ-Pitman–Yor process with $\epsilon = 0.01$ is 0.00570. Thus, the truncation level of 50 and running time of 30 s give us an approximation for $F(1/3)$ whose accuracy is slightly better than the ϵ-Pitman–Yor process with $\epsilon = 0.01$.

4 | POSTERIOR INFERENCE SCHEME

In this section, we develop a posterior inference scheme for the truncated two-parameter Poisson–Dirichlet process and illustrate its usage in the Pitman–Yor process mixture models. Call $\mathbf{Y} = \{Y_1, \ldots, Y_n\}$ a sample of the Pitman–Yor process if

$$(Y_i|\tilde{P}) \overset{iid}{\sim} \tilde{P}, \quad i = 1, \ldots, n, \quad \tilde{P} \sim \tilde{\mathcal{P}}_\infty, \tag{13}$$

then the joint distribution of $Y_1, \ldots, Y_n$ can be characterized by the Pólya urn scheme (6). To approximate the posterior distribution of model (13), we replace the Pitman–Yor process prior with the truncated two-parameter Poisson–Dirichlet process. Then we consider a sequence of observations $\mathbf{Y} = \{Y_1, \ldots, Y_n\}$ from P_N, that is,

$$(Y_i|P_N) \overset{iid}{\sim} P_N = \sum_{j=1}^{N} p_j \delta_{K_j}(.) + e_N \delta_{K_0}(.), \quad i = 1, \ldots, n, \quad P_N \sim \mathcal{P}_N. \tag{14}$$

Denote by n_j the number of observations taking the value of K_j, such that $\sum_{j=0}^{N} n_j = n$, then the observations can be written as $\mathbf{Y} = (n_1, \ldots, n_N, n_0)$. To make inference about the random weights

$\mathbf{P} := (p_1,\ldots,p_N,e_N)$, we need to sample from the posterior $\mathbb{P}(\mathbf{P}|\mathbf{Y})$. But to determine $\mathbf{P}$, it is sufficient to find out the values of $(J_1,R_1,\ldots,R_{N-1},{}^{(N)}\tau_1)$ in the representations (8) and (9). To facilitate the posterior inference scheme, we re-express the representations as

$$p_i = \frac{Z^{-1/\alpha}\Pi^{-1}}{Z^{-1/\alpha}\Pi^{-1}\Sigma + {}^{(N)}\tau_1}\prod_{j=1}^{i-1}R_j, \quad i=1,\ldots,N, \quad \text{and} \quad e_N = \frac{{}^{(N)}\tau_1}{Z^{-1/\alpha}\Pi^{-1}\Sigma + {}^{(N)}\tau_1}, \tag{15}$$

where $\Pi := R_1\ldots R_{N-1}$, $\Sigma := 1 + R_1 + R_1R_2 + \cdots + R_1R_2\ldots R_{N-1}$, and the joint density of $(R_1,\ldots,R_{N-1},Z,{}^{(N)}\tau_1)$ is (see the proof of Algorithm 1)

$$f(r_1,\ldots,r_{N-1},z,y) := \frac{\Gamma(\theta+1)\Gamma(1-\alpha)^{\theta/\alpha}}{\Gamma(\theta/\alpha+1)}\frac{e^{-z}\alpha^{N-1}z^{\theta/\alpha+N-1}r_1^{(\theta-1)+\alpha}\ldots r_{N-1}^{(\theta-1)+(N-1)\alpha}}{(\Sigma+z^{1/\alpha}\Pi y)^{\theta}}f_{\alpha,z^{-1/\alpha}}(y)dydzdr_1\ldots dr_{N-1}.$$

On the other hand, the likelihood of the truncated two-parameter Poisson–Dirichlet process is given by $\mathbb{P}(\mathbf{Y}|\mathbf{P}) \sim \text{Multi}(n_1,\ldots,n_N,n_0;p_1,\ldots,p_N,e_N)$. Thus, we can express the posterior of the truncated two-parameter Poisson–Dirichlet process as

$$\mathbb{P}(r_1,\ldots,r_{N-1},z,y|\mathbf{Y}) \propto \frac{z^{n_0/\alpha}\Pi^{n_0}r_1^{n_2+\cdots+n_N}\ldots r_{N-1}^{n_N}y^{n_0}}{(\Sigma+z^{1/\alpha}\Pi y)^{n}}f(r_1,\ldots,r_{N-1},z,y). \tag{16}$$

To sample from (16), we develop a blocked Gibbs sampler to draw iteratively from $\mathbb{P}(r_1,\ldots,r_{N-1}|z,y,\mathbf{Y})$ and $\mathbb{P}(z,y|r_1,\ldots,r_{N-1},\mathbf{Y})$. We use the Hamiltonian Monte Carlo method (HMC; see Neal, 2011, see also Sect. 7.2 of Caron & Fox, 2017 for a similar application in posterior sampling) to sample from

$$\mathbb{P}(r_1,\ldots,r_{N-1}|z,y,\mathbf{Y}) \propto \Pi^{n_0}(\Sigma+z^{1/\alpha}\Pi y)^{-(n+\theta)}\prod_{i=1}^{N-1}r_i^{(n_{i+1}+\cdots+n_N)+(\theta-1)+i\alpha}.$$

The HMC method is based on the computation of the gradient of the log-posterior, which, after the change of variables $\mathcal{R}_i := \tan(\pi(r_i-0.5))$, is denoted by $D(\mathcal{R}_1,\ldots,\mathcal{R}_{N-1}) := \nabla\log(\mathbb{P}(\mathcal{R}_1,\ldots,\mathcal{R}_{N-1}|z,y,\mathbf{Y}))$. The derivation of the gradient is straightforward, we attach the details in the Appendix S1.

Algorithm 2 (Hamiltonian Monte Carlo). *Let $L \geq 1$ be the number of leapfrog steps and $\epsilon > 0$ be the step size. The HMC algorithm for $\mathbb{P}(r_1,\ldots,r_{N-1}|z,y,\mathbf{Y})$ is given as follows.*

1. *Load the current values of $(r_1,\ldots,r_{N-1})$, set $\tilde{W}^{(0)} \leftarrow (\tan(\pi(r_1-0.5)),\ldots,\tan(\pi(r_{N-1}-0.5)))$.*
2. *Sample from a multivariate normal distribution $p \leftarrow \mathcal{N}(0,I_{N-1})$, set $\tilde{p}^{(0)} \leftarrow p + (\epsilon/2)D(\tilde{W}^{(0)})$.*
3. *For $l = 1,\ldots,L-1$, set $\tilde{W}^{(l)} \leftarrow \tilde{W}^{(l-1)} + \epsilon\tilde{p}^{(l-1)}$ and $\tilde{p}^{(l)} \leftarrow \tilde{p}^{(l-1)} + \epsilon D(\tilde{W}^{(l)})$.*
4. *Set $\tilde{W} \leftarrow \tilde{W}^{(L-1)} + \epsilon\tilde{p}^{(L-1)}$, $\tilde{p} \leftarrow -\{\tilde{p}^{(L-1)} + (\epsilon/2)D(\tilde{W})\}$ and $(\tilde{r}_1,\ldots,\tilde{r}_{N-1}) \leftarrow 0.5 + (1/\pi)\arctan(\tilde{W})$.*

5. *Sample from a uniform distribution* $U \leftarrow U[0,1]$, *if*

$$U \leq \frac{\mathbb{P}(\tilde{r}_1,\ldots,\tilde{r}_{N-1}|z,y,\mathbf{Y})}{\mathbb{P}(r_1,\ldots,r_{N-1}|z,y,\mathbf{Y})}\exp\left(-\frac{1}{2}\sum_{i=1}^{N-1}(\tilde{p}_i{}^2-p_i^2)\right),$$

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

$$\mathbb{P}(z,y|r_1,\ldots,r_{N-1},\mathbf{Y}) \propto z^{(n_0+\theta)/\alpha+N-1}e^{-z}y^{n_0}f_{\alpha,z^{-1/\alpha}}(y)(\Sigma+z^{1/\alpha}\Pi y)^{-(n+\theta)}.$$

Algorithm 3 (Metropolis–Hastings). *The Metropolis–Hastings algorithm for* $\mathbb{P}(z,y|r_1,\ldots,r_{N-1},\mathbf{Y})$ *is given as follows.*

1. *Load the current values of* z,y.
2. *Sample from a gamma distribution* $\tilde{z} \leftarrow gamma((n_0+\theta)/\alpha+N,1)$.
3. *Sample from a truncated stable process* $\tilde{y} \leftarrow \tilde{z}^{-1/\alpha} \times AlgorithmTS(\alpha,\Gamma(1-\alpha)\tilde{z})$.
4. *Sample from a uniform distribution* $U \leftarrow U[0,1]$. *If*

$$U \leq \frac{\tilde{y}^{n_0}(\Sigma+\tilde{z}^{1/\alpha}\Pi\tilde{y})^{-(n+\theta)}}{y^{n_0}(\Sigma+z^{1/\alpha}\Pi y)^{-(n+\theta)}},$$

accept the candidates and output $(\tilde{z},\tilde{y})$. *Otherwise, output* (z,y).

We can now formulate the blocked Gibbs sampler as follows.

Algorithm 4 (Blocked Gibbs sampler). *Posterior inference for the truncated two-parameter Poisson–Dirichlet process.*

1. *Update* $R_1,\ldots,R_{N-1}$: *Sample from* $\mathbb{P}(r_1,\ldots,r_{N-1}|z,y,\mathbf{Y})$ *using Algorithm 2.*
2. *Update* $Z,{}^{(N)}\tau_1$: *Sample from* $\mathbb{P}(z,y|r_1,\ldots,r_{N-1},\mathbf{Y})$ *using Algorithm 3.*
3. *Update* $p_1,\ldots,p_N,e_N$: *Use the representations (15).*

We obtain the posterior values of $(p_1,\ldots,p_N,e_N)$ by running Algorithm 4 iteratively. If we input an empty set of observations, that is, $n=0$, the posterior (16) reverts to the prior distribution, and the block Gibbs sampler draws directly from the prior. This gives us the MCMC algorithm for the truncated two-parameter Poisson–Dirichlet process prior. Its performance has been demonstrated in Table 1. As explained in the previous section, this algorithm is particularly useful when θ is large.

Note that we have used fixed α and θ in the blocked Gibbs sampler. It is also possible to put priors on these parameters and estimate them in the posterior inference scheme. The existing literature has considered various priors. For example, Lijoi et al. (2008) suggested the uniform priors with finite and fixed support; Jara et al. (2010) used a pair of beta and truncated normal priors; Carmona et al. (2019) applied a pair of beta and truncated gamma priors. Other methods include the usage of fixed values (see Ishwaran & James, 2001), empirical Bayes rule specification (see Lijoi et al., 2007), comparison between different combinations of fixed values (see Navarrete et al., 2008) and learning these parameters (see Cereda et al., 2023; Favaro et al., 2009).

Next, we adapt the truncated two-parameter Poisson–Dirichlet process into a Pitman–Yor process mixture model. We replace the Pitman–Yor process prior $\tilde{\mathcal{P}}_\infty$ in model (5) by a truncated two-parameter Poisson–Dirichlet process $\mathcal{P}_N$. To achieve an efficient Markov chain Monte Carlo sampling scheme, we also recast the model completely in terms of random variables. Then model (5) is approximated by

$$\begin{aligned}(X_i|\mathbf{Z},\mathbf{K}) &\sim \mathcal{K}(X_i|Z_{K_i}), \quad i=1,\ldots,n,\\ (K_i|P_N) &\overset{iid}{\sim} P_N=\sum_{j=1}^{N} p_j\delta_j(.)+e_N\delta_0(.),\\ P_N &\sim \mathcal{P}_N,\\ \mathbf{Z} &\sim \pi(\mathbf{Z}),\end{aligned} \tag{17}$$

where $\mathbf{K}=(K_1,\ldots,K_n)$ are the classifiers that relate the random variables $\mathbf{Z}=(Z_1,\ldots,Z_n)$ to the latent variables Y_i, that is, $Y_i=Z_{K_i}$. Using the blocked Gibbs sampler for the truncated two-parameter Poisson–Dirichlet process, we can devise a posterior inference scheme for model (17). The sampler draws iteratively from the conditional distributions $\mathbb{P}(\mathbf{P}|\mathbf{K})$, $\mathbb{P}(\mathbf{Z}|\mathbf{K},\mathbf{X})$ and $\mathbb{P}(\mathbf{K}|\mathbf{P},\mathbf{Z},\mathbf{X})$, thus producing the posterior values from $\mathbb{P}(\mathbf{P},\mathbf{Z},\mathbf{K}|\mathbf{X})$.

We illustrate the usage of the posterior inference scheme within a normal mean mixture model. The model is in the format of (5) with $\mathcal{K}(X_i|Y_i)\sim\mathcal{N}(X_i|\mu_i,\sigma_i)$. We choose the priors $\sigma^{-1}\sim \mathrm{Ga}(a_0,b_0)$ and $\mu|\sigma\sim\mathcal{N}(\theta_\mu,5\sigma)$. Using (17), we approximate the normal mean mixture model by

$$\begin{aligned}(X_i|\mathbf{Z},\mathbf{K}) &\sim \mathcal{N}(\mu_{K_i},\sigma_{K_i}), \quad i=1,\ldots,n,\\ (K_i|P_N) &\overset{iid}{\sim} P_N=\sum_{j=1}^{N} p_j\delta_j(.)+e_N\delta_0(.),\\ P_N &\sim \mathcal{P}_N,\\ (\mu_j|\sigma_j) &\sim \mathcal{N}(\theta_\mu,5\sigma_j), \quad j=0,\ldots,N,\\ (\sigma_j^{-1}|a_0,b_0) &\sim \mathrm{Ga}(a_0,b_0), \quad j=0,\ldots,N.\end{aligned} \tag{18}$$

We develop a blocked Gibbs sampler for the posterior of model (18) in the following algorithm. Note that the derivation of Step 2, 3, and 4 of the algorithm can be found in Ishwaran and James (2002).

Algorithm 5 (Normal mean mixture model). *Posterior inference scheme for model (18).*

1. *Update* $(p_1,\ldots,p_N,e_N)$*: Sample from* $\mathbb{P}(\boldsymbol{P}|\boldsymbol{K})$ *using Algorithm 4.*
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*

$$\mu_j\leftarrow\mathcal{N}(\theta_j^*,\sigma_{Z_j}^*),\ \text{where}\ \theta_j^*=\sigma_{Z_j}^*(\theta_\mu/(5\sigma_j)+\sum_{\{i:K_i=K_j^*\}}X_i/\sigma_j),\ \sigma_{Z_j}^*=(n_j/\sigma_j+1/(5\sigma_j))^{-1},$$

and n_j *is the number of times* K_j^* *occurs in* $\boldsymbol{K}$*. Also, for each* $j\in\boldsymbol{K}-\{K_1^*,\ldots,K_m^*\}$*, simulate* $\mu_j\leftarrow\mathcal{N}(\theta_\mu,5\sigma_j)$.

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

$$(\sigma_j^{-1}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{K}) \sim \text{Ga}(a_0 + n_j/2, b_0 + \sum_{\{i:K_i=K_j^*\}} (X_i-\mu_j)^2/2).$$

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

4. *Update* $\boldsymbol{K}$*: For* $i = 1,\ldots,n$*, draw* K_i *from*

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

where

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

The outcome of Algorithm 5 can be used to estimate the predictive density for a new observation. We denote by $f(X_{n+1}|\mathbf{X})$ the predictive density of X_{n+1} conditioned on the current observations $\mathbf{X} = (X_1,\ldots,X_n)$ and Y_{n+1} the latent variables of this new observation, then

$$f(X_{n+1}|\mathbf{X}) = \int \mathcal{K}(X_{n+1}|Y_{n+1})d\mathbb{P}(Y_{n+1}|\mathbf{X}) = \iint \mathcal{K}(X_{n+1}|Y_{n+1})d\mathbb{P}(Y_{n+1}|P_N)d\mathbb{P}(P_N|\mathbf{X}).$$

Based on a truncated two-parameter Poisson–Dirichlet process prior, the inner integral can be expressed as

$$\int \mathcal{K}(X_{n+1}|Y_{n+1})d\mathbb{P}(Y_{n+1}|P_N) = \sum_{j=1}^{N} p_j \mathcal{K}(X_{n+1}|Y_j) + e_N \mathcal{K}(X_{n+1}|Y_0). \tag{19}$$

Thus, the predictive density $f(X_{n+1}|\mathbf{X})$ can be estimated by averaging (19) over the posterior values from different iterations.

5 | NUMERICAL IMPLEMENTATIONS

In this section, we carry out a simulation study to examine the effectiveness of the posterior inference scheme in Section 4 and compare it with other inference methods. For the latter, we will consider the truncated stick-breaking process in Ishwaran and James (2001), the dependent and independent slice-efficient samplers proposed by Kalli et al. (2011) and the importance conditional sampler (ICS) developed by Canale et al. (2022). The slice sampler was first proposed by Walker (2007) as an exact sampler for the Dirichlet process mixture models that avoids introducing the truncation error. The slice-efficient sampler is a modification which solves the problems of slow mixing and generating too many random weights that arise from the slice sampler. Even

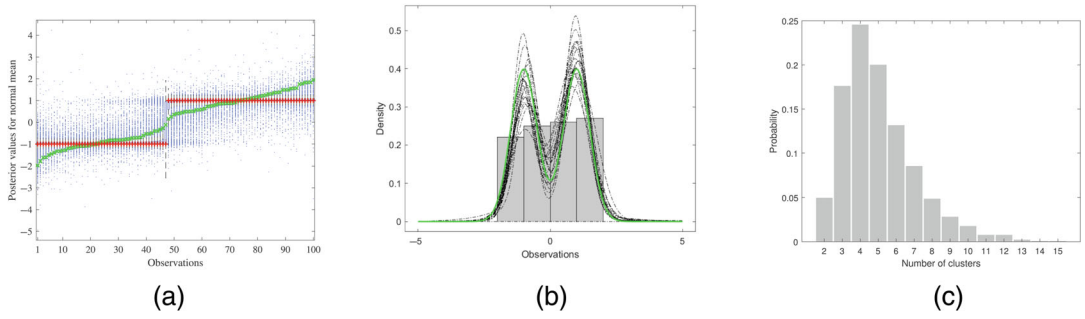

FIGURE 5 Posterior inference of the bimodal mixture with $n = 100$ observations using Algorithm 5. The parameter values are $\alpha = 0.3$, $\theta = 1$ and $N = 100$. The plots are based on a sample size of 10000 iterations following an initial burn-in of 10,000 iterations. (a) Posterior mean values. The observations X_i and true values for μ_X are denoted by green cross and red plus, respectively. (b) Twenty-five randomly selected predictive densities evaluated over the same partition. The observations are presented by the histogram, and the true density curve of the bimodal mixture is plotted in green. (c) Proportions of the number of occupied clusters.

though, when it comes to a Pitman–Yor process prior, the slice-efficient sampler could still be extremely slow or even unfeasible due to the huge number of random variables generated, in particular when the discount parameter α is large. See Canale et al. (2022) for further discussion. To facilitate the posterior inference of a Pitman–Yor process mixture model, Canale et al. (2022) proposed the importance conditional sampler, which combines the appealing features of both conditional and marginal methods while avoiding their weaknesses. We will use these methods to analyze the synthetic data generated by a bimodal mixture and a leptokurtic mixture. The bimodal mixture assumes that $f(x_i) = 0.5\mathcal{N}(-1, 0.5^2) + 0.5\mathcal{N}(1, 0.5^2)$, and the leptokurtic mixture assumes that $f(x_i) = (2/3)\mathcal{N}(0, 1) + (1/3)\mathcal{N}(0.3, 0.25^2)$. We will also consider the inference problem of the galaxy velocity data, which is widely used in Bayesian non-parametric statistics. The analysis is carried out on MATLAB 2023a on a 64-bit Windows desktop with an Intel i9-10900 processor and 64GB RAM.

The first part of our numerical study is to check the posterior inference results. We draw $n = 100$ independent samples from the bimodal mixture and use Algorithm 5 to estimate the model posterior. We choose the concentration parameter $\theta = 1$, discount parameter $\alpha = 0.3$, truncation level $N = 100$ and hyperparameters $a_0 = 2$, $b_0 = 1$, $\theta_\mu = 0$. In the HMC step, we use the leapfrog steps $L = 10$ and adjust the step size to obtain an acceptance rate of around 0.6. The numerical results are presented in Figure 5. From the figures we can see that the predictive density induces two modes for the observations at -1 and 1. We also count the number of occupied clusters in each iteration and record their proportions. The results suggest that the posterior distribution has at least two clusters, and it is unlikely to have more than 12 clusters. Clearly, the number of occupied clusters is overestimated. To achieve a more concentrated posterior estimation, we try a smaller concentration parameter $\theta = 0.3$. The numerical results are given in Figure 6. We can see that the posterior induces a similar predictive density as before but with fewer occupied clusters.

In the second part of the numerical study, we compare the performance of the different posterior inference schemes. We draw $n = 100$ independent samples from the leptokurtic mixture and analyze the data using different methods. The numerical results are recorded in Table 3 and Table 4. In these tables, "Slice (Dep.)" and "Slice (Ind.)" represent the dependent and independent slice-efficient sampler, and "Truncation" stands for the truncated stick-breaking process.

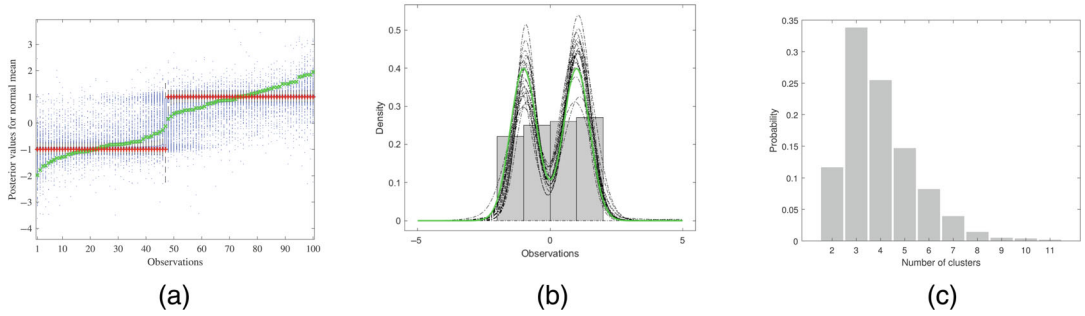

FIGURE 6 Posterior inference of the bimodal mixture with $n = 100$ observations using Algorithm 5. The parameter values are $\alpha = 0.3$, $\theta = 0.3$ and $N = 100$. (a) Posterior mean values; (b) Predictive densities; (c) Proportions of occupied clusters.

TABLE 3 Posterior of the leptokurtic mixture, sample size $n = 100$, $\alpha = 0.3$, $\theta = 1.0$, and the running time is 300 s for all the posterior inference schemes.

	$\hat{\tau}_K$	$\hat{\tau}_D$	$\overline{K}$	$\overline{D}$	E_K	E_D	SSE	SSAE	Sample size
Slice (Dep.)	26.8223	3.7556	14.1661	256.1072	274	1982	56.5466	78.6271	15,643
Slice (Ind.)	43.2989	3.9441	14.2197	255.9956	156	1255	56.7274	78.4549	12,338
ICS	9.8518	2.9189	9.5206	256.3662	9864	32792	60.3621	78.8972	199,070
Truncation ($N = 50$)	11.1042	2.6034	8.9575	256.8672	8898	32002	50.2342	79.4275	176,050
Truncation ($N = 100$)	10.8349	2.5288	8.9672	256.8431	6317	25807	50.1550	79.3786	131,480
Algorithm 5 ($N = 50$)	9.0631	2.3133	8.8879	256.8071	1725	5648	50.2038	79.4123	34,852
Algorithm 5 ($N = 100$)	8.4802	1.9134	9.3181	256.7904	582	2606	49.3115	79.4962	12,213

Abbreviations: and D, number of occupied deviance; E, effective sample size is rounded to the nearest integer; K, the number of occupied clusters; SSAE, the sum of standardized absolute errors; SSE, sum of squared errors; $\hat{\tau}$, the estimated IAT.

TABLE 4 Posterior of the leptokurtic mixture, sample size $n = 100$, $\alpha = 0.3$, $\theta = 5.0$, and the running time is 300 s for all the posterior inference schemes.

	$\hat{\tau}_K$	$\hat{\tau}_D$	$\overline{K}$	$\overline{D}$	E_K	E_D	SSE	SSAE	Sample size
Slice (Dep.)	14.0635	2.1074	25.6977	256.1467	132	588	53.1386	77.5859	3419
Slice (Ind.)	16.4477	1.6796	27.9333	256.4582	62	476	53.2785	77.2834	2296
ICS	5.5329	1.7067	16.3845	256.1058	8311	26,493	55.2525	78.2405	99,010
Truncation ($N = 50$)	7.1361	1.3725	15.8804	256.4368	13697	64,508	45.8796	79.3409	175,480
Truncation ($N = 100$)	6.5592	1.2977	16.6150	256.4570	12415	45,583	45.7911	79.3852	124,310
Algorithm 5 ($N = 50$)	5.1625	1.3058	16.0237	256.4124	2794	11,029	45.2089	79.6978	30,443
Algorithm 5 ($N = 100$)	5.0519	1.2208	16.4793	256.3884	1139	3940	45.7102	79.4084	12,092

Abbreviations: SSAE, the sum of standardized absolute errors; SSE, sum of squared errors; $\hat{\tau}$, the estimated IAT.

We would like to remind the reader of the remarkable BNPmix package developed by Corradin et al. (2021), which contains the ICS and slice-efficient samplers. On the other hand, the "Truncation" method is obtained by replacing Step 1 of Algorithm 5 with the posterior of the truncated stick-breaking process, which can be found in Suppl. A of Jara et al. (2010). We also include a concise derivation of the posterior in the supplementary materials. The running time is set to 300 s for all the posterior inference schemes to achieve a fair comparison.

The inferential quality is monitored by four quantities: the number of occupied clusters, the deviance of the estimated density, the sum of squared errors (SSE) and the sum of standardized absolute errors (SSAE). For each iteration r of the blocked Gibbs sampler, denote by $K^{(r)}$ the number of occupied clusters and $n_j^{(r)}$ the size of each occupied cluster, such that $\sum_{j=0}^{K^{(r)}} n_j^{(r)} = n$. The deviance is a function of all estimated parameters defined as

$$D^{(r)} := -2\sum_{i=1}^{n} \log\left(\sum_{j=0}^{K^{(r)}} \frac{n_j^{(r)}}{n} \mathcal{K}(X_i|Y_j^{(r)})\right).$$

The SSE denotes the sum of the squared differences between the observation x_i and the predictive mean $\mathbb{E}(X_i|\text{data})$. The SSAE stands for the sum of the standardized error $|x_i - \mathbb{E}(X_i|\text{data})|/\sqrt{\text{Var}(X_i|\text{data})}$. These quantities have been used in the previous comparison studies of Neal (2000), Papaspiliopoulos and Roberts (2008), Fall and Barat (2012), Kalli et al. (2011), Canale et al. (2022), and Argiento et al. (2016). The algorithm efficiency can be evaluated by calculating the integrated autocorrelation time (IAT) and effective sample size (ESS) of $K^{(r)}$ and $D^{(r)}$. The IAT of a variable is defined by Sokal (1997) as $\tau := 0.5 + \sum_{l=1}^{\infty} \rho_l$, 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 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 cut-off point to be selected by the user. We will use the same cut-off point as Kalli et al. (2011), that is, $C := \min\{l : |\hat{\rho}_l| < 2/\sqrt{M}\}$, 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 $H_0 : \rho_l = 0$. See Kalli et al. (2011) for more details. On the other hand, the ESS measures how many posterior samples are effective. Due to the autocorrelation, the number of effective samples would be smaller than the length of the Markov chain, and a higher ESS implies a better sequence of posterior samples. In practice, the ESS can be computed by the CODA package. We refer the reader to Plummer et al. (2006) and Canale et al. (2022) for more details.

From the IAT results, we can see that the slice samplers are less efficient than the other methods. Algorithm 5 and the ICS method have similar efficiency, and they are more efficient than the truncated stick-breaking method with a reduction in IAT of 10 to 20 per cent. On the other hand, we notice that Algorithm 5 provides the smallest SSE. It is known that the SSE is an index favoring complex models and leading to better values when the data set is over-fitted (see, e.g., Argiento et al., 2016), and Algorithm 5 is preferable according to the SSE criterion. Finally, we notice that Algorithm 5 generates fewer samples in the given time compared to the ICS and truncated stick-breaking process. This is caused by the simulation of the truncated stable process, which costs a lot of time. See Dassios et al. (2020) for a further discussion about the simulation efficiency. We conduct another numerical experiment based on the bimodal mixture and provide the results in the Appendix S1. The findings from the bimodal mixture are similar to those before. Specifically, Algorithm 5 is more efficient than the truncated stick-breaking process method and leads to the lowest SSE. In addition, we use Algorithm 5 to analyze the galaxy velocity data to

demonstrate its usage in real-world datasets. The posterior mean density estimation results are provided in the Appendix S1.

6 | DISCUSSION

In this paper, we have studied the finite approximation of the Pitman–Yor process by truncating its two-parameter Poisson–Dirichlet representation. We call the approximation process a truncated two-parameter Poisson–Dirichlet process. We have developed two simulation algorithms for the truncated two-parameter Poisson–Dirichlet process and used them to estimate the functionals of the Pitman–Yor process. The simulation results in Section 3 show that, for the same running time, our method provides a better approximation to the functionals compared to the truncated stick-breaking process. We have also adapted the truncated two-parameter Poisson–Dirichlet process into a Pitman–Yor process mixture model and designed the posterior inference scheme for the model. Numerical implementations suggest that our posterior inference scheme is more efficient than the truncated stick-breaking process method and leads to a lower SSE than the other methods.

The construction of the truncated two-parameter Poisson–Dirichlet process is based on the ranked sequence of the stick-breaking random weights. Thus, the existing research about the truncation error of the stick-breaking process provides an upper bound for the truncation error of our process. In this paper, we have compared the truncation error of different approximation methods. However, we did not give a rule for selecting the truncation level, except for simply looking at the expectation of the truncation error. In fact, our construction method needs to determine all the random weights simultaneously. Thus, a predefined truncation level is required. But it is still possible to use a random truncation level by looking at the ratio between two consecutive random weights. For example, the existing literature has considered a random stopping rule M in terms of $J_M/\sum_{i=1}^{M} J_i < \epsilon$ for a completely random measure (see, e.g., Arbel & Prünster, 2017; Gelfand & Kottas, 2002). We can extend this method to the current work by adding a condition $(R_1 \ldots R_{M-1})/(1 + R_1 + \cdots + R_1 \ldots R_{M-1}) < \epsilon$ into Step 1 of Algorithm 1, such a random stopping rule truncates the sequence when the newly sampled random weight is small enough compared to the previous ones.

We find the application of the truncated two-parameter Poisson–Dirichlet process in the approximation of the Pitman–Yor process mixture models. Numerical implementations suggest a reasonable estimation quality by using the approximation. We also find that the simulation of the truncated stable process slows down the posterior inference scheme. Specifically, Step 3 of Algorithm 3 simulates a truncated stable process at time $t := \Gamma(1-\alpha)\tilde{z}$. When t is large, the exact simulation algorithm becomes less efficient. One potential improvement could be to split the time into $t = \sum_{i=1}^{\lfloor t \rfloor} 1 + (t - \lfloor t \rfloor)$, where $\lfloor t \rfloor$ denotes the largest integer smaller than t. Then we can simulate $\lfloor t \rfloor$ number of truncated stable processes, each at time 1, parallelly, and a truncated stable process at time $t - \lfloor t \rfloor$. Their summation gives us a sample of the truncated stable process at time t. Alternatively, we could replace the exact simulation algorithm with a nonexact but faster sampler. This problem will be further investigated in future work.

ACKNOWLEDGMENTS

We are grateful to the editor and two reviewers for giving us extremely helpful comments and suggestions.

ORCID

Junyi Zhang https://orcid.org/0000-0001-8986-6588

REFERENCES

Al Labadi, L., & Zarepour, M. (2014). On simulations from the two-parameter Poisson-Dirichlet process and the normalized inverse-Gaussian process. *Sankhya A*, *76*, 158–176.

Arbel, J., De Blasi, P., & Prünster, I. (2019). Stochastic approximations to the Pitman-Yor process. *Bayesian Analysis*, *14*, 1201–1219.

Arbel, J., & Prünster, I. (2017). A moment-matching Ferguson & Klass algorithm. *Statistics and Computing*, *27*, 3–17.

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

Bassetti, F., Casarin, R., & Leisen, F. (2014). Beta-product dependent Pitman-Yor processes for Bayesian inference. *Journal of Econometrics*, *180*, 49–72.

Campbell, T., Huggins, J. H., How, J. P., & Broderick, T. (2019). Truncated random measures. *Bernoulli*, *25*, 1256–1288.

Canale, A., Corradin, R., & Nipoti, B. (2022). Importance conditional sampling for Pitman-Yor mixtures. *Statistics and Computing*, *32*, Paper No. 40, 18.

Carmona, C., Nieto-Barajas, L., & Canale, A. (2019). Model-based approach for household clustering with mixed scale variables. *Advances in Data Analysis and Classification*, *13*, 559–583.

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

Cereda, G., Corradi, F., & Viscardi, C. (2023). Learning the two parameters of the Poisson-Dirichlet distribution with a forensic application. *Scandinavian Journal of Statistics*, *50*(1), 120–141.

Corradin, R., Canale, A., & Nipoti, B. (2021). BNPmix: An R package for Bayesian nonparametric modeling via Pitman-Yor mixtures. *Journal of Statistical Software*, *100*, 1–33.

Dassios, A., Lim, J. W., & Qu, Y. (2020). Exact simulation of a truncated Lévy subordinator. *ACM Transactions on Modeling and Computer Simulation*, *30*(Art. 17), 26.

Dassios, A., & Zhang, J. (2021). Exact simulation of two-parameter Poisson-Dirichlet random variables. *Electronic Journal of Probability*, *26*, Paper No. 5, 20.

Fall, M. D., & Barat, É. (2012). *Gibbs sampling methods for Pitman-Yor mixture models*. INRIA.

Favaro, S., Lijoi, A., Mena, R. H., & Prünster, I. (2009). Bayesian non-parametric inference for species variety with a two-parameter Poisson-Dirichlet process prior. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *71*, 993–1008.

Gelfand, A. E., & Kottas, A. (2002). A computational approach for full nonparametric Bayesian inference under Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, *11*, 289–305.

Ghosal, S., & van der Vaart, A. (2017). *Fundamentals of nonparametric Bayesian inference Cambridge series in statistical and probabilistic mathematics* (Vol. *44*). Cambridge University Press.

Handa, K. (2009). The two-parameter Poisson-Dirichlet point process. *Bernoulli*, *15*, 1082–1116.

Horváth, G., Horváth, I., Almousa, S. A.-D., & Telek, M. (2020). Numerical inverse Laplace transformation using concentrated matrix exponential distributions. *Performance Evaluation*, *137*, 102067.

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

Ishwaran, H., & James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, *96*, 161–173.

Ishwaran, H., & James, L. F. (2002). Approximate Dirichlet process computing in finite normal mixtures: Smoothing and prior information. *Journal of Computational and Graphical Statistics*, *11*, 508–532.

James, L. F., Lijoi, A., & Prünster, I. (2010). On the posterior distribution of classes of random means. *Bernoulli*, *16*, 155–180.

Jara, A., Lesaffre, E., De Iorio, M., & Quintana, F. (2010). Bayesian semiparametric inference for multivariate doubly-interval-censored data. *The Annals of Applied Statistics*, *4*, 2126–2149.

Kalli, M., Griffin, J. E., & Walker, S. G. (2011). Slice sampling mixture models. *Statistics and Computing*, *21*, 93–105.

Lijoi, A., Mena, R. H., & Prünster, I. (2007). A Bayesian nonparametric method for prediction in EST analysis. *BMC Bioinformatics*, *8*, 1–10.

Lijoi, A., Mena, R. H., & Prünster, I. (2008). A Bayesian nonparametric approach for comparing clustering structures in EST libraries. *Journal of Computational Biology*, *15*, 1315–1327.

McCloskey, J. W. (1965). *A model for the distribution of individuals by species in an environment*. Department of Statistics, Michigan State University.

Muliere, P., & Tardella, L. (1998). Approximating distributions of random functionals of Ferguson-Dirichlet priors. *The Canadian Journal of Statistics*, *26*, 283–297.

Navarrete, C., Quintana, F. A., & Müller, P. (2008). Some issues in nonparametric Bayesian modeling using species sampling models. *Statistical Modelling*, *8*, 3–21.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, *9*, 249–265.

Neal, R. M. (2011). *MCMC using Hamiltonian dynamics*. In S. Brooks, A. Gelman, G. Jones & X.-L. Meng (Eds.), *Handbook of Markov chain Monte Carlo Chapman & Hall/CRC Handbooks of Modern Statistical Methods* (pp. 113–162). CRC Press.

Ni, Y., Müller, P., Zhu, Y., & Ji, Y. (2018). Heterogeneous reciprocal graphical models. *Biometrics*, *74*, 606–615.

Papaspiliopoulos, O., & Roberts, G. O. (2008). Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, *95*, 169–186.

Perman, M., Pitman, J., & Yor, M. (1992). Size-biased sampling of Poisson point processes and excursions. *Probability Theory and Related Fields*, *92*, 21–39.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, *102*, 145–158.

Pitman, J. (2006). *Combinatorial stochastic processes Lecture notes in mathematics* (Vol. *1875*). Springer-Verlag.

Pitman, J., & Yor, M. (1992). Arcsine laws and interval partitions derived from a stable subordinator. *Proceedings of the London Mathematical Society. Third Series*, *65*, 326–356.

Pitman, J., & Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, *25*, 855–900.

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). CODA: Convergence diagnosis and output analysis for MCMC. *R News*, *6*, 7–11.

Sokal, A. (1997). *Monte Carlo methods in statistical mechanics: Foundations and new algorithms*. In C. DeWitt-Morette, P. Cartier, & A. Folacci (Eds.), *Functional integration (Cargèse, 1996) NATO Advanced Science Institutes Series B: Physics* (Vol. *361*, pp. 131–192). Plenum.

Sudderth, E., & Jordan, M. (2008). Shared segmentation of natural scenes using dependent Pitman-Yor processes. *Advances in Neural Information Processing Systems*, *21*, 1585–1592.

Walker, S. G. (2007). Sampling the Dirichlet mixture model with slices. *Communications in Statistics. Simulation and Computation*, *36*, 45–54.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Zhang, J., & Dassios, A. (2023). Truncated two-parameter Poisson–Dirichlet approximation for Pitman–Yor process hierarchical models. *Scandinavian Journal of Statistics*, 1–22. https://doi.org/10.1111/sjos.12688