Gibbs sampling for mixtures in order of appearance: the ordered allocation sampler

Gibbs sampling for mixtures in order of appearance: the ordered allocation sampler

Pierpaolo De Blasi
Collegio Carlo Alberto and ESOMAS Department,
University of Torino, Italy
pierpaolo.deblasi@unito.it

María F. Gil–Leyva
Department of Probability and Statistics, IIMAS–UNAM, México
marifer@sigma.iimas.unam.mx

Abstract

Gibbs sampling methods are standard tools to perform posterior inference for mixture models. These have been broadly classified into two categories: marginal and conditional methods. While conditional samplers are more widely applicable than marginal ones, they may suffer from slow mixing in infinite mixtures, where some form of truncation, either deterministic or random, is required. In mixtures with random number of components, the exploration of parameter spaces of different dimensions can also be challenging. We tackle these issues by expressing the mixture components in the random order of appearance in an exchangeable sequence directed by the mixing distribution. We derive a sampler that is straightforward to implement for mixing distributions with tractable size-biased ordered weights, and that can be readily adapted to mixture models for which marginal samplers are not available. In infinite mixtures, no form of truncation is necessary. As for finite mixtures with random dimension, a simple updating of the number of components is obtained by a blocking argument, thus, easing challenges found in trans-dimensional moves via Metropolis-Hastings steps. Additionally, sampling occurs in the space of ordered partitions with blocks labelled in the least element order, which endows the sampler with good mixing properties. The performance of the proposed algorithm is evaluated in a simulation study.

Keywords: Dirichlet process; Pitman-Yor process; size-biased permutations; stick-breaking construction; species sampling models.

1 Introduction

Mixture models represent one of the most successful applications of Bayesian methods. Bayesian inference proceeds by placing a prior on the mixing distribution, whose atoms and their sizes represent the mixture component parameters and the weights, respectively. An important issue is that the number of components is rarely known in advance. The nonparametric approach consists in modelling the mixing distribution with infinitely many support points, and infer the number of components through the number of groups observed in the data (e.g. Escobar and West; 1995). Another alternative is to assign a prior to this unknown quantity (cf. Richardson and Green; 1997; Miller and Harrison; 2018). Posterior inference for mixture models is customarily based on Gibbs sampling methods which have been broadly classified into two categories: *marginal* and *conditional* samplers. Marginal methods (Escobar and West; 1995; Neal; 2000) are termed this way because they partially integrate out the mixing distribution and exploit the generalized Pólya urn scheme representation (Blackwell and MacQueen; 1973; Pitman; 2006) of the prediction rule of a sample from the mixing distribution. By doing so marginal samplers avoid dealing with a potentially infinite or random model dimension. They are well suited for models such as the Dirichlet (Ferguson; 1973; Escobar and West; 1995), the Pitman-Yor (Pitman and Yor; 1997; Ishwaran and James; 2001) and mixtures of finite mixtures (Miller and Harrison; 2018). However, they are challenging to adapt to mixing priors without a tractable prediction rule. In alternative, one can use conditional methods which include the mixing distribution and update it as a component of the sampler. While being more widely applicable they bring some issues in their design and implementation. In infinite mixture models, some sort of truncation either deterministic or random is necessary to avoid dealing with infinitely many mixture components. Finite dimensional approximations of the mixing prior were proposed in (Ishwaran and James; 2001). Of random truncation type are the exact conditional samplers derived by Walker (2007); Kalli et al. (2011) and Papaspiliopoulos and Roberts (2008). It has been observed that they require Metropolis-Hastings steps that swap components' labels so to speed up mixing (Porteous et al.; 2006; Papaspiliopoulos and Roberts; 2008). As for mixtures with a random number of components, the main challenge of conditional samplers is the need to explore parameter spaces of different dimensions. The standard method is the reversible jump MCMC algorithm (Richardson and Green; 1997) but this can be difficult to implement.

In this paper we contribute both methodologically and computationally to mitigate these issues by developing a novel conditional Gibbs sampling method named the *ordered allocation sampler*. The sampler works with the mixture components in the random order in which they are discovered. To derive it we use in depth the theory of species sampling models set forth in Pitman (1995, 1996a,b) where, in particular, it is established that the law of the weights in order of appearance corresponds to the distribution of the weights that is invariant under size-biased permutations. This one admits a simple stick-breaking representation for the Dirichlet and the Pitman-Yor processes. Working with this specific rearrangement of mixture components allows us to exploit a (conditional) prediction rule in the sampler. Thus, it bears similarities with marginal methods such as the fact that mixing takes place in the space of partitions and not in the space of cluster's labels as it occurs in other conditional samplers (Porteous et al.; 2006). Empirical studies confirm that this endows our sampler with nice mixing properties. A second major advantage of our proposal is that, since n data points can not be generated from more than n distinct components, at most the sampler needs to update the first n components in order of appearance. This is especially relevant for infinite mixture models as it avoids truncation. In particular, for Pitman-Yor processes with slowly decaying weights, our sampler proves to be very

convenient computationally wise. A third important consequence is that marginalization over the weights in order of appearance yields exactly the *exchangeable partition probability function* (EPPF Pitman; 1996b). For mixtures with random dimension, this translates to a simple way of updating the number of components without resorting to a reversible jump step. Finally, as other conditional methods, the ordered allocation sampler allows direct inference on the mixing distribution and can be adapted to a wide range of mixing priors.

The rest of the paper is organized as follows. In Section 2 we provide background theory on species sampling priors in mixture models. It will set the stage for the ordered allocation sampler. In Section 3 we first derive the sampler for models with tractable size-biased permuted weights, and later we show how to adapt it when the law of this arrangement of the weights is not available in explicit form. In Section 4 we illustrate the performance of our sampler with well-known real and simulated datasets. Some concluding remarks and discussion points are brought in Section 5. Proofs, technical details and additional illustrations are in the Appendix.

2 Species sampling models

In Bayesian mixture models we model exchangeable data, $(y_i) = (y_i)_{i=1}^n$, taking values in a Borel space, $(\mathbb{Y}, \mathcal{B}(\mathbb{Y}))$, as conditionally independent and identically distributed (iid) from

$$Q(y) = \int_{\mathbb{X}} g(y \mid x) P(\mathrm{d}x) = \sum_{j=1}^{m} p_j \, g(y \mid x_j), \tag{1}$$

where $g(\cdot \mid x)$ is a density for each x, and the mixing distribution, $P = \sum_{j=1}^m p_j \delta_{x_j}$, is an almost surely discrete random probability measure over the Borel parameter space $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$. *Species sampling models*, introduced and studied by Pitman (1996b), constitute a very general class of random probability measures that provide a convenient prior specification for the mixing distribution P. In species sampling models, $P = \sum_{j=1}^m p_j \delta_{x_j}$, the atoms, $(x_j) = (x_1, \ldots, x_m)$, are iid from a diffuse distribution, ν, over $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$, and are independent of $(m, (p_j))$. The weights $(p_j) = (p_1, \ldots, p_m)$ are positive random variables with $\sum_{j=1}^m p_j = 1$ almost surely, and the number of support points, m, can be finite, infinite or random. A sequence, $(\theta_i) = (\theta_i)_{i=1}^\infty$, is a *species sampling sequence* driven by P if it is exchangeable and the almost sure limit of the empirical distributions, $P = \lim_{n\to\infty} n^{-1} \sum_{i=1}^n \delta_{\theta_i}$, is a species sampling model. By de Finetti's theorem, the latter is equivalent to the existence of a species sampling model, P, such that given P, (θ_i) are conditionally iid according to P (cf. Theorem 1.1 and Proposition 1.4 of Kallenberg; 2005). The laws of (θ_i) and P determine each other, and both are fully determined by that of $(m, (p_j))$, and the diffuse distribution, ν. A key aspect to note is that the law of P is invariant under weights permutations. That is, $P = \sum_{j=1}^m p_j \delta_{x_j}$ is equal in distribution to $\sum_{j=1}^m p_{\rho(j)} \delta_{x_j}$ for every permutation ρ of $\{1, \ldots, m\}$, which means that working with an ordering of the weights or another does not change the mixing prior. This is reflected through the so called exchangeable partition probability function (EPPF), given by

$$\pi(n_1, \ldots, n_k) = \sum_{(j_1, \ldots, j_k)} \mathbb{E}\Big(\prod_{i=1}^{k} p_{j_i}^{n_i} \Big), \tag{2}$$

where the sum ranges over all k-tupples of distinct positive integers, and $p_j = 0$ for $j > m$. In fact, $\pi(n_1, \ldots, n_k)$ describes the probability that a sample, $(\theta_1, \ldots, \theta_n)$, of size $n = \sum_{j=1}^k n_j$,

exhibits exactly k distinct values, with corresponding frequencies, $n_1, \ldots, n_k$ (Pitman; 1996b, 2006). Whenever $\pi(n_1, \ldots, n_k)$ can be computed in closed form, the prediction rule, $\mathbb{P}[\theta_{i+1} \in \cdot \mid \theta_1, \ldots, \theta_i]$, of (θ_i) is available and can be described in terms of a generalized Pólya urn scheme (cf Blackwell and MacQueen; 1973).

The invariance under permutations of $\sum_{j=1}^m p_{\rho(j)}\delta_{x_j}$, as well as the complexity of computing the unordered sum in (2), have motivated the study of weights permutations that simplify the analysis. An ordering of the weights of paramount importance is the *size-biased permutation*, $(\tilde{p}_j) = (\tilde{p}_1, \ldots, \tilde{p}_m)$, given by $\tilde{p}_j = p_{\alpha_j}$, and $(\alpha_j) = (\alpha_1, \ldots, \alpha_m)$ defined by

$$\begin{aligned} \mathbb{P}[\alpha_1 = j \mid (p_j)] &= p_j, \\ \mathbb{P}[\alpha_l = j \mid (p_j), \alpha_1, \ldots, \alpha_{l-1}] &= \frac{p_j}{1 - \sum_{i=1}^{l-1} p_{\alpha_i}} \mathbb{1}_{\{j \notin \{\alpha_1, \ldots, \alpha_{l-1}\}\}}, \quad 2 \leq l \leq m. \end{aligned} \tag{3}$$

In other words, (α_j) and $(\tilde{p}_j)$ are sampled without replacement from $\{1, \ldots, m\}$ and (p_j), respectively, with probabilities (p_j). By construction the distribution of $(\tilde{p}_j)$ is *invariant under size-biased permutations*. As shown by Pitman (1995, 1996a), the EPPF (2) can be computed through

$$\pi(n_1, \ldots, n_k) = \mathbb{E}\left[\prod_{j=1}^{k} \tilde{p}_j^{n_j - 1} \prod_{j=1}^{k-1} \left(1 - \sum_{l=1}^{j} \tilde{p}_l\right)\right], \tag{4}$$

hence, if the distribution of $(\tilde{p}_j)$ is available, it becomes easier to compute $\pi(n_1, \ldots, n_k)$. Another advantage of working with size-biased permutations is that $\tilde{p}_j$ coincides with the long-run proportion of indexes i such that $\theta_i = \tilde{x}_j$, where $\tilde{x}_j$ is the jth distinct value to appear in (θ_i). Furthermore, the conditional law of (θ_i) given $(\tilde{x}_j) = (\tilde{x}_1, \ldots, \tilde{x}_m)$ and $(\tilde{p}_j)$ admits a simple prediction rule as detailed next.

Theorem 1. *Let $P = \sum_{j=1}^m p_j \delta_{x_j}$ be a species sampling model over the Borel space $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$ and let $(\theta_i) = (\theta_i)_{i=1}^{\infty}$ be a sequence with values in $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$. Define the jth distinct value to appear in (θ_i) through $\tilde{x}_j = \theta_{M_j}$, where $M_j = \min\{i > M_{j-1} : \theta_i \notin \{\tilde{x}_1, \ldots, \tilde{x}_{j-1}\}\}$, for $j \geq 2$ and $M_1 = 1$. Then (θ_i) is an species sampling sequence driven by P if and only if the following hold:*

i. (θ_i) exhibits m distinct values, $(\tilde{x}_j) = (\tilde{x}_1, \ldots, \tilde{x}_m)$, in order of appearance, and $(\tilde{x}_j)$ are iid from ν. Furthermore, for $j \leq m$, $\tilde{x}_j = x_{\alpha_j}$ where (α_j) satisfies (3).

ii. The almost sure limits

$$\tilde{p}_j = \lim_{n \to \infty} \frac{|\{i \leq n : \theta_i = \tilde{x}_j\}|}{n}, \quad j \leq m,$$

exist, $\tilde{p}_j > 0$, $\sum_{j=1}^m \tilde{p}_j = 1$ almost surely, and $(\tilde{p}_j) = (\tilde{p}_1, \ldots, \tilde{p}_m)$ is invariant under size-biased permutations. Moreover, $(\tilde{p}_j)$ is given by $\tilde{p}_j = p_{\alpha_j}$, with (α_j) as in 2.i.

iii. $\theta_1 = \tilde{x}_1$, and the conditional prediction rule of (θ_i) given $(\tilde{p}_j)$ and $(\tilde{x}_j)$ is

$$\mathbb{P}[\theta_{i+1} \in \cdot \mid (\tilde{p}_j), (\tilde{x}_j), \theta_1, \ldots, \theta_i] = \sum_{j=1}^{k_i} \tilde{p}_j \delta_{\tilde{x}_j} + \left(1 - \sum_{j=1}^{k_i} \tilde{p}_j\right) \delta_{\tilde{x}_{k_i+1}},$$

for every $i \geq 1$, where k_i is the number of distinct values in $(\theta_1, \ldots, \theta_i)$.

iv. m, $(\tilde{p}_j)$, (p_j) *and* (α_j) *are independent of elements in* $(\tilde{x}_j)$.

Theorem 1 is based on theory laid down in Pitman (1995, 1996b), nonetheless, we provide a self-contained proof in Appendix A, due to the crucial role it plays in the derivation of the new sampler.

The canonical example of species sampling models in Bayesian nonparametric statistics is the Dirichlet process (Ferguson; 1973). It has $m = \infty$ support points and its size-biased permuted weights, $(\tilde{p}_j)$, admit the stick-breaking representation

$$\tilde{p}_1 = v_1, \quad \tilde{p}_j = v_j \prod_{i=1}^{j-1}(1 - v_i), \quad j \geq 2, \tag{5}$$

where $(v_j) = (v_j)_{j=1}^{\infty}$ are iid from the Beta distribution $\mathsf{Be}(1, \theta)$ (Sethuraman; 1994). The Dirichlet model can be generalized to the two-parameter (σ, θ)-model (Pitman; 2006) which features size-biased permuted weights as in (5) with independent $v_j \sim \mathsf{Be}(1-\sigma, \theta+j\sigma)$ according to one of the following two regimes:

a) $\sigma \in [0, 1)$ and $\theta > -\sigma$. In this case $m = \infty$ and the species sampling model P has been named the Pitman-Yor process by Ishwaran and James (2001) after Pitman and Yor (1997). Evidently the choice $\sigma = 0$ reduces to a Dirichlet process.

b) Given $m \in \mathbb{N}$, $\sigma = -\gamma < 0$, and $\theta = m\gamma$. In agreement with the notation we have established m stands for the number of support points of P. It turns out that the law of $(\tilde{p}_j)$ corresponds to that of the size-biased permutation of symmetric Dirichlet weights, $(p_1, \ldots, p_m) \sim \mathsf{Dir}(\gamma, \ldots, \gamma)$ (Pitman; 1996a). When γ is fixed and m is random P belongs to the class of Gibbs-type priors (see De Blasi et al.; 2015, for a recent review), while the allied mixture model corresponds to the mixture of finite mixtures of Miller and Harrison (2018).

Another type of species sampling models for which a stick-breaking characterization of size-biased weights is available are homogeneous normalized random measures with independent increments (cf. Regazzini et al.; 2003; Favaro et al.; 2016). Unfortunately, such characterization remains elusive for most species sampling models used in mixture modelling, examples are finite dimensional approximations of the Pitman-Yor process (Ishwaran and James; 2001), the Geometric process (Fuentes-García et al.; 2010), the probit stick-breaking process (Rodríguez and Dunson; 2011) and exchangeable stick-breaking processes studied by Gil–Leyva and Mena (2021). For all these species sampling priors the weights can be defined in terms of a stick-breaking decomposition, $p_j = v_j \prod_{l=1}^{j-1}(1 - v_l)$, for some sequence of random variables (v_j) with values in $[0, 1]$, yet (p_j) is not invariant under size-biased permutations.

3 The ordered allocation sampler

As mentioned in Section 2, in mixture models data points $(y_i) = (y_i)_{i=1}^{n}$ are treated as conditionally iid from a random density Q as in (1). Whenever the mixing distribution, P, is a species sampling model we can equivalently assume $y_i \mid \theta_i \sim g(\cdot \mid \theta_i)$, independently for $i \leq n$, where (θ_i) is a species sampling sequence driven by P. In this setting, marginal samplers integrate out

P and exploit the exchangeability of (θ_i) as well as the prediction rule, $\mathbb{P}[\theta_{i+1} \in \cdot \mid \theta_1, \ldots, \theta_i]$, to derive an algorithm for posterior inference (cf. Neal; 2000; Favaro and Teh; 2013; Miller and Harrison; 2018). Instead, conditional samplers (e.g. Ishwaran and James; 2001; Papaspiliopoulos and Roberts; 2008; Kalli et al.; 2011) include the mixing distribution, P, and update its atoms, (x_j), and weights, (p_j), as components of the sampler. The ordered allocation sampler is a conditional sampler as it includes the mixing distribution, P, however similarly to marginal samplers it relies on a prediction rule for species sampling sequences. Explicitly, motivated by Theorem 1 we work with the atoms, $(\tilde{x}_j)$, and weights, $(\tilde{p}_j)$, of P in the order in which they were discovered by (θ_i). As commonly done in other samplers, we augment the model with latent allocation variables that identify each observation, y_i, with the mixture component it was sampled from. Here, in accordance with the order of appearance we introduce what we call *ordered allocation variables*, $(d_i) = (d_i)_{i=1}^n$, given by $d_i = j$ if and only if y_i was sampled from $g(\cdot \mid \tilde{x}_j)$, i.e. $\theta_i = \tilde{x}_j$. Thus, $\theta_i = \tilde{x}_{d_i}$, and $y_i \mid ((\tilde{x}_j), d_i) \sim g(\cdot \mid \tilde{x}_{d_i})$, independently for $i \leq n$. If k_i denotes the number of distinct values in $(\theta_1, \ldots, \theta_i)$ then k_i coincides with $\max\{d_1, \ldots, d_i\}$, and d_{i+1} necessarily takes a value in $\{1, \ldots, k_i + 1\}$. More precisely, *iii* of Theorem 1 allows us to compute

$$d_1 = 1, \quad d_{i+1} \mid (\tilde{p}_j), d_1, \ldots, d_i \sim \sum_{j=1}^{k_i} \tilde{p}_j \delta_j + \left(1 - \sum_{j=1}^{k_i} \tilde{p}_j\right) \delta_{k_i+1}, \tag{6}$$

for $i \leq n$, independently of elements in $(\tilde{x}_j)$ (see also *iv* of Theorem 1). This yields the augmented likelihood

$$\mathbb{p}[(y_i), (d_i) \mid (\tilde{p}_j), (\tilde{x}_j)] = \prod_{j=1}^{k_n} \tilde{p}_j^{n_j - 1} \left(1 - \sum_{l=1}^{j-1} \tilde{p}_l\right) \prod_{i \in D_j} g(y_i \mid \tilde{x}_j) \mathbb{1}_{\mathcal{D}}, \tag{7}$$

where $k_n = \max\{d_1, \ldots, d_n\}$, $D_j = \{i \leq n : d_i = j\}$, $n_j = |D_j|$, and $\mathcal{D}$ is the event that $\{D_1, \ldots, D_{k_n}\}$ is a partition of $\{1, \ldots, n\}$ with blocks in the *least element order*, in particular $D_j \neq \emptyset$, for $j \leq k_n$, and $\min(D_1) < \min(D_2) < \cdots < \min(D_{k_n})$. The full conditional distributions required at each iteration of the sampler are proportional to the product of (7) times the prior distributions, $\mathbb{p}[(\tilde{x}_j)]$ and $\mathbb{p}[(\tilde{p}_j)]$, of the atoms and weights of P in order of appearance. We first derive the ordered allocation sampler for those species sampling mixing distributions where the prior of $(\tilde{p}_j)$ can be modelled directly as is the case of the (σ, θ)-model. Latter we explain how to adapt the sampler for the more general case where the law of $(\tilde{p}_j)$ is not available.

3.1 Ordered allocation sampler for size-biased weights

Updating of the ordered allocation variables (d_i):

$$\mathbb{p}[\, d_i = d \mid \cdots] \propto \tilde{p}_d \, g(y_i \mid \tilde{x}_d) \times \prod_{j=1}^{k_n} \tilde{p}_j^{-1} \left(1 - \sum_{l=1}^{j-1} \tilde{p}_l\right) \mathbb{1}_{\mathcal{D}}. \tag{8}$$

This is the fundamentally novel part of the algorithm. Differently from other conditional samplers, the allocation variables (d_i) can not be updated independently of each other for two main reasons: (i) $k_n = \max\{d_1, \ldots, d_n\}$ might change as a consequence of an update in d_i, and (ii) the least element order of $D_1, \ldots, D_{k_n}$ must be preserved, as specified by the indicator $\mathbb{1}_{\mathcal{D}}$.

Instead, the updating of (d_i) resembles the way marginal algorithms update allocation variables (cf. Neal; 2000) in the sense that we will update one d_i at a time by conditioning on the current value of the remaining ordered allocation variables. To do so, we first identify the set, $\mathcal{D}_i$, of *admissible moves* for d_i, which contains all positive integers d for which the event $\mathcal{D}$ remains true after setting $d_i = d$. That is, for $d \in \mathbb{N}$, define $k_n^{(d)} = \max\{k_{-i}, d\}$, where $k_{-i} = \max\{d_l : l \neq i\}$, and add $d \in \mathcal{D}_i$ if, under the assumption $d_i = d$, the sets $D_j = \{l \leq n : d_l = j\}$ are non-empty, for $j \leq k_n^{(d)}$, and satisfy $\min(D_1) < \min(D_2) < \cdots < \min\left(D_{k_n^{(d)}}\right)$. An example that illustrates how to determine $\mathcal{D}_i$ is available in Appendix D. With this notation at hand we can rewrite (8):

$$\mathbb{p}[\, d_i = d \mid \cdots] \propto \tilde{p}_d \, g(y_i \mid \tilde{x}_d) \times \prod_{j=1}^{k_n^{(d)}} \tilde{p}_j^{-1} \left(1 - \sum_{l=1}^{j-1} \tilde{p}_l\right) \mathbb{1}_{\{d \in \mathcal{D}_i\}}. \tag{9}$$

Next, we need to weight the admissible moves according to (9). To this aim, note that for each $d \in \mathcal{D}_i$, either $k_n^{(d)} = k_{-i}$ or $k_n^{(d)} = k_{-i} + 1$. This means that we can divide (9) by $\prod_{j=1}^{k_{-i}} \tilde{p}_j^{-1}(1 - \sum_{l=1}^{j-1} \tilde{p}_l)$ and obtain

$$\mathbb{p}[\, d_i = d \mid \cdots] \propto \begin{cases} \tilde{p}_d g(y_i \mid \tilde{x}_d) & \text{if } k_n^{(d)} = k_{-i}, \\ \left(1 - \sum_{l=1}^{k_{-i}} \tilde{p}_l\right) g(y_i \mid \tilde{x}_d) & \text{if } k_n^{(d)} = k_{-i} + 1, \end{cases} \tag{10}$$

for $d \in \mathcal{D}_i$, and $\mathbb{p}[\, d_i = d \mid \cdots] = 0$, for $d \notin \mathcal{D}_i$. Once we have identified $\mathcal{D}_i$, sampling from (10) is straight-forward, as its support is $\mathcal{D}_i \subset \{1, \ldots, n\}$.

Updating of component parameters in order of appearance $(\tilde{x}_j)$:

$$\mathbb{p}[\, \tilde{x}_j \mid \cdots] \propto \prod_{i \in D_j} g(y_i \mid \tilde{x}_j) \nu(\tilde{x}_j). \tag{11}$$

Since $D_j = \emptyset$, for $j > k_n$, we simply sample $\tilde{x}_j \sim \nu$ from its prior distribution. For $j \leq k_n$, sampling from (11) is easy if g and ν form a conjugate pair. Otherwise, the problem of updating the non-empty components parameters is identical as in conditional samplers and some marginal ones. The advantage with respect to conditional algorithms is that the occupied component parameters are precisely the first k_n in the sequence $(\tilde{x}_j)$.

Updating of size-biased weights $(\tilde{p}_j)$:

$$\mathbb{p}[\, (\tilde{p}_j) \mid \cdots] \propto \prod_{j=1}^{k_n} \tilde{p}_j^{n_j - 1} \left(1 - \sum_{l=1}^{j-1} \tilde{p}_l\right) \times \mathbb{p}[(\tilde{p}_j)]. \tag{12}$$

If the stick-breaking representation (5) is available, we can update $(\tilde{p}_j)$ via sampling (v_j) from its full conditional. Noting that $1 - \sum_{l=1}^{k} \tilde{p}_l = \prod_{l=1}^{k}(1 - v_l)$ for each $k \geq 1$, we find

$$\mathbb{p}[\, (v_j) \mid \cdots] \propto \left[\prod_{j=1}^{k_n} v_j^{n_j - 1}(1 - v_j)^{\sum_{l>j} n_l}\right] \times \mathbb{p}[(v_j)].$$

For example, for the (σ,θ)-model we know that apriori $v_j \sim \mathsf{Be}(1-\sigma, \theta+j\sigma)$, independently, according to one of the two regimes (a) or (b) spelled out in Section 2. In this case, we update $v_j \sim \mathsf{Be}(n_j - \sigma, \theta + j\sigma + \sum_{l>j} n_l)$, independently for $j \leq k_n$, and we sample $v_j \sim \mathsf{Be}(1-\sigma, \theta+j\sigma)$, for $j > k_n$, as apriori.

Before we move on, there are two points worth highlighting concerning the updating of $(\tilde{p}_j)$. The first one is that while the stick-breaking decomposition simplifies this step it is not a requirement, what is needed is a posterior characterization of the weights in order of appearance. The Pitman-Yor multinomial process studied by Lijoi et al. (2020) illustrates this point. The second crucial remark is that sampling $\tilde{p}_j$ and $\tilde{x}_j$, for $j > k_n$, is needed for only a few j as required by the occupation of new components when updating (d_i). Being that $d_i \leq n$, at most we will require to update $\tilde{p}_j$ and $\tilde{x}_j$, for $j \leq \min\{n, m\}$. This is specially relevant for infinite mixture models as it assures the sampler unfolds in a finite dimensional space even when the model dimension, m, is infinite. In fact, if the model dimension is deterministic, the ordered allocation sampler is practically identical for finite and infinite mixture models. The case of random m is treated next.

Updating the model dimension m:

If the model dimension is random, our proposal here is to update m, $(\tilde{p}_j)$ and the non occupied component parameters, $(\tilde{x}_j)_{j>k_n} = (\tilde{x}_{k_n+1}, \ldots, \tilde{x}_m)$, as a block from

$$\mathbb{p}[\, m, (\tilde{p}_j), (\tilde{x}_j)_{j>k_n} \mid \cdots] \propto \prod_{j=1}^{k_n} \tilde{p}_j^{n_j-1} \left(1 - \sum_{l=1}^{j-1} \tilde{p}_l\right) \prod_{j=k_n+1}^{m} \nu(\tilde{x}_j) \times \mathbb{p}[\,(\tilde{p}_j) \mid m]\mathbb{p}[m]. \quad (13)$$

Here we keep "$\cdots$" to denote all random terms other than m, $(\tilde{p}_j)$ and $(\tilde{x}_j)_{j>k_n}$. We first sample m from its *marginal*, i.e. (13) after integrating over $(\tilde{p}_j)$ and $(\tilde{x}_j)_{j>k_n}$:

$$\mathbb{p}[\, m \mid \cdots] \propto \mathbb{E}\left[\prod\nolimits_{j=1}^{k_n} \tilde{p}_j^{n_j-1} \left(1 - \sum\nolimits_{l=1}^{j-1} \tilde{p}_l\right) \,\middle|\, m \right] \mathbb{p}(m). \quad (14)$$

The expectation is taken with respect to the conditional distribution of $(\tilde{p}_j)$ given m and treating $k_n, n_1, \ldots, n_{k_n}$ as constants. In particular, since $\sum_{j=1}^{m} \tilde{p}_j = 1$, (14) equals zero for $m < k_n$. Taking this into account and recognizing, in the conditional expectation, the EPPF of the species sampling model given m, cf. (4), we obtain

$$\mathbb{p}[\, m \mid \cdots] \propto \pi(n_1, \ldots, n_{k_n} \mid m)\mathbb{p}[m]\mathbb{1}_{\{k_n \leq m\}}. \quad (15)$$

This is a remarkably simple expression for the updating of the model dimension as it only requires the conditional EPPF given m. In Appendix B we provide an example on how to update m for mixtures of finite mixtures and for the choice of $\mathbb{p}[m]$ detailed by Gnedin (2010). After updating m, we sample $(\tilde{p}_j)$ and $(\tilde{x}_j)_{j>k_n}$, conditioning on m. Thus $(\tilde{p}_j)$ is sampled from (12) as detailed before, and the $m - k_n$ non occupied component parameters, $\tilde{x}_{k_n+1}, \ldots, \tilde{x}_m$, from the prior ν, cf. (11). Note that the blocking argument is remarkably simple when compared with the Metropolis-Hasting steps of the reversible jump MCMC algorithm.

3.2 Ordered allocation sampler for non size-biased weights

In this section we adapt the ordered allocation sampler to species sampling priors that do not enjoy an explicit characterization of the size-biased weights $(\tilde{p}_j)$. This makes our sampler applicable to mixture models for which marginal samplers are not available, so increasing substantially its scope (examples can be found at the end of Section 2). To this aim recall that $(\tilde{p}_j)$ is a rearrangement of the weights in any arbitrary order, (p_j), i.e. $\tilde{p}_j = p_{\alpha_j}$, where (α_j) is sampled without replacement from $\{1, \dots, m\}$ with probabilities (p_j), as defined in (3). The key idea is to include (α_j) as part of the sampler. As we will see, this augmentation yields a conditional sampler that inherits the advantages of the algorithm in Section 3.1 without requiring a closed-form expression of $(\tilde{p}_j)$ or the EPPF. As for the component parameters in order of appearance, $(\tilde{x}_j)$, we will continue to model them directly, as i.i.d. from ν, independently of (p_j) and (α_j), cf. *iv* in Theorem 1. Thus, instead of (7), we work with the augmented likelihood

$$\mathbb{p}[\,(y_i),(d_i) \mid (p_j),(\alpha_j),(\tilde{x}_j)\,] = \prod_{j=1}^{k_n} p_{\alpha_j}^{n_j-1}\left(1-\sum_{l=1}^{j-1} p_{\alpha_l}\right)\prod_{i\in D_j} g(y_i \mid \tilde{x}_j)\mathbb{1}_{\mathcal{D}}. \tag{16}$$

It is straightforward to see that the updating of the ordered allocation variables (d_i) and the component parameters $(\tilde{x}_j)$ remain identical. Hence, we will only explain how to update the weights in order of appearance via (p_j) and (α_j), as well as the model dimension, m, whenever this quantity is random.

Updating of $(\tilde{p}_j)$ through (p_j) and (α_j):

From (3) and (16) we find

$$\mathbb{p}[\,(p_j),(\alpha_j) \mid \cdots] \propto \prod_{j=1}^{k_n} p_{\alpha_j}^{n_j} \times \prod_{j=k_n+1}^{m} p_{\alpha_j}\left(1-\sum_{l=1}^{j-1} p_{\alpha_l}\right)^{-1}\mathbb{1}_{\mathcal{A}} \times \mathbb{p}[(p_j)], \tag{17}$$

where $\mathcal{A}$ is the event that $\alpha_i \neq \alpha_j$ for every $i \neq j$. In this part, as a notational device, we keep "$\cdots$" to denote all random terms other than $(p_j), (\alpha_j)$. Also, we distinguish the indexes of the occupied components, $(\alpha_j)_{j\leq k_n}$, from the remaining ones, $(\alpha_j)_{j>k_n}$. The key idea to attain simple updating steps is to sample $(\alpha_j)_{j\leq k_n}$ from its full conditional, which can be expressed as a *weighted permutation* of k_n indexes, and separately (p_j) and $(\alpha_j)_{j>k_n}$ as a block.

We first focus on the updating of $(\alpha_j)_{j\leq k_n}$. From (17) we get

$$\mathbb{p}[\,(\alpha_j)_{j\leq k_n} \mid (\alpha_j)_{j>k_n},(p_j),\cdots] \propto \prod_{j=1}^{k_n} p_{\alpha_j}^{n_j}\mathbb{1}_{\mathcal{A}}, \tag{18}$$

after noting that $\prod_{j=k_n+1}^{m}(1-\sum_{l=1}^{j-1} p_{\alpha_l})^{-1}$ is a constant with respect to $(\alpha_j)_{j\leq k_n}$, because $1-\sum_{l=1}^{j-1} p_{\alpha_l} = \sum_{l=j}^{m} p_{\alpha_l}$. The event $\mathcal{A}$ indicates that this is about sampling from a weighted permutation ρ of the k_n integers corresponding to current values of $(\alpha_j)_{j\leq k_n}$. Namely, we sample ρ from

$$\pi(\rho) = \frac{1}{Z}\prod_{j=1}^{k_n} w_{j,\rho(j)}\,, \quad \rho \in \mathcal{S},$$

where $w_{j,l} = p_{\alpha_l}^{n_j}$, Z is the normalizing constant and $\mathcal{S}$ is the space of permutations of $\{1, \dots, k_n\}$. Afterwards we simply apply ρ to the indexes of the current value of $(\alpha_j)_{j \le k_n}$ so to obtain the updated value $(\alpha_{\rho(j)})_{j \le k_n}$. Now, to sample ρ from π we follow Zanella (2020) by adopting a Metropolis–Hastings scheme using a *locally-balanced* informed proposal distribution, cf. Example 3 therein. For the reader's convenience, we recall briefly how it works. Let $N(\rho)$ be the neighborhood of $\rho \in \mathcal{S}$ given by all permutations obtained by switching two indexes, (i.e. $\rho^* \in N(\rho)$ if and only if there exist $i \neq j$ such that $\rho^*(i) = \rho(j)$, $\rho^*(j) = \rho(i)$ and $\rho^*(l) = \rho(l)$ for all $l \notin \{i, j\}$). Instead of using a random walk scheme, consisting in proposing a new value of ρ, say ρ^*, uniformly over $N(\rho)$, and accepting it with probability $a(\rho, \rho^*) = \min\{1, \pi(\rho^*)/\pi(\rho)\}$, we bias the proposal towards high probability regions of the target. To do so, we set the proposal distribution to be $Q(\rho, \rho^*) = \sqrt{\pi(\rho^*)}\mathbb{1}_{N(\rho)}/Z(\rho)$, where $Z(\rho) = \sum_{z \in N(\rho)} \sqrt{\pi(z)}$ is the normalizing constant. Then the new value, ρ^*, is accepted with probability $a(\rho, \rho^*) = \min\left\{1, \frac{\pi(\rho^*)Q(\rho^*, \rho)}{\pi(\rho)Q(\rho, \rho^*)}\right\}$. In the simulation study we initialized ρ as the identity function over $\mathcal{S}$ and performed k_n Metropolis-Hastings steps at each iteration. As explained by Zanella (2020) the appeal of considering a locally-balanced proposal, such as Q, is that it is roughly π-reversible when $\mathcal{S}$ is large with respect to $N(\rho)$, thus the acceptance probability $a(\rho, \rho^*)$ tends to be high. Otherwise, if k_n is small, one can opt to sample exactly from (18) by enumerating all possible permutations of $(\alpha_j)_{j \le k_n}$.

As for the updating of (p_j) and $(\alpha_j)_{j > k_n}$, we first sample (p_j) from the conditional distribution $\mathbb{p}[\,(p_j) \mid (\alpha_j)_{j \le k_n}, \cdots]$ obtained from $\mathbb{p}[\,(p_j), (\alpha_j)_{j > k_n} \mid (\alpha_j)_{j \le k_n}, \cdots]$ after integrating over $(\alpha_j)_{j > k_n}$. We get

$$\mathbb{p}[\,(p_j) \mid (\alpha_j)_{j \le k_n}, \cdots] \propto \prod_{j=1}^{\overline{\alpha}} p_j^{r_j} \times \mathbb{p}[(p_j)],$$

where $\overline{\alpha} = \max\{\alpha_j : j \le k_n\}$, and $r_j = \sum_{l=1}^{k_n} n_l \mathbb{1}_{\{\alpha_l = j\}}$, that is $r_j = n_l$ if and only if there exist $l \le k_n$ such that $\alpha_l = j$ and $r_j = 0$ otherwise. When $p_j = v_j \prod_{l=1}^{j-1}(1 - v_l)$, we can update (p_j) via sampling (v_j) from

$$\mathbb{p}[\,(v_j) \mid (\alpha_j)_{j \le k_n}, \cdots] \propto \prod_{j=1}^{\overline{\alpha}} v_j^{r_j}(1 - v_j)^{\sum_{l>j} r_j} \times \mathbb{p}[(v_j)].$$

For instance, if a priori $v_j \sim \mathsf{Be}(a_j, b_j)$, independently for $j \ge 1$, then a posteriori $v_j \sim \mathsf{Be}(r_j + a_j, \sum_{l>j} r_l + b_j)$ for $j \le \overline{\alpha}$ and $v_j \sim \mathsf{Be}(a_j, b_j)$, for $j > \overline{\alpha}$. Further examples on how to update (p_j) can be found in Appendix C. After updating (p_j) we sample $(\alpha_j)_{j > k_n}$ from

$$\mathbb{p}[\,(\alpha_j)_{j > k_n} \mid (\alpha_j)_{j \le k_n}, (p_j), \cdots] = \prod_{j=k_n+1}^{m} p_{\alpha_j} \left(1 - \sum_{l=1}^{j-1} p_{\alpha_l}\right)^{-1} \mathbb{1}_{\mathcal{A}}. \tag{19}$$

That is, $\alpha_{k_n+1}, \alpha_{k_n+2}, \dots$ are sampled without replacement from $\{j \le m : \; j \notin (\alpha_j)_{j \le k_n}\}$ with probabilities proportional to $\{p_j : \; j \notin (\alpha_j)_{j \le k_n}\}$. This can be achieved by sampling sequentially as a priori, cf. (3).

This way of updating (α_j), although theoretically valid, has the disadvantage that switches among indexes in $(\alpha_j)_{j \le k_n}$ and indexes in $(\alpha_j)_{j > k_n}$ only occur when k_n changes as a consequence of an update in (d_i). To facilitate the mixing one can include the following *acceleration step* after updating $(\alpha_j)_{j \le k_n}$ from (18) and before updating $(\alpha_j)_{j > k_n}$ from (19). We suggest to sample

each α_j with $j \leq k_n$ from

$$\mathbb{p}[\alpha_j \mid (p_j), (\alpha_l)_{l \leq k_n, l \neq j}, \cdots] \propto p_{\alpha_j}^{n_j} \mathbb{1}_{\mathcal{A}}, \tag{20}$$

i.e. conditioning on the current values of α_l, for $l \leq k_n$ and $l \neq j$, with $(\alpha_j)_{j>k_n}$ integrated out. Hence, the indicator $\mathbb{1}_{\mathcal{A}}$ above only dictates $\alpha_j \neq \alpha_l$, with $l \leq k_n$. If the number of components is finite, the support of (20) consists of $m - k_n - 1$ positive integers and sampling directly from this distribution is trivial. Otherwise, when $m = \infty$, we can treat $(\alpha_j)_{j>k_n}$ as a latent variable with distribution as in (19), and update the pair $(\alpha_j, \alpha_{k_n+1})$ from

$$\mathbb{p}[(\alpha_j, \alpha_{k_n+1}) \mid (p_j), (\alpha_l)_{l \notin \{j, k_n+1\}}, \cdots] \propto p_{\alpha_j}^{n_j} p_{\alpha_{k_n+1}} \left(1 - \sum_{l=1}^{k_n} p_{\alpha_l}\right)^{-1} \mathbb{1}_{\mathcal{A}}. \tag{21}$$

In practice, it is enough to sample α_{k_n+1} from $\mathbb{p}[\alpha_{k_n+1} \mid \alpha_1, \ldots, \alpha_{k_n}, (p_j), \cdots]$ as in (3) and later either leave $(\alpha_j, \alpha_{k_n+1})$ unchanged or switch the values of α_j and α_{k_n+1}, with probabilities determined by (21). It is worth emphasizing that this procedure has to be repeated for all $j \leq k_n$, and that each time we discard α_{k_n+1} because it is only playing the role of an auxiliary variable to draw samples from (20).

Similarly as with the sampler in Section 3.1, this sampler unfolds in a finite dimensional space even when the model dimension is infinite. In general, we will only need to update $\tilde{x}_j$ and α_j, for $j > k_n$, when required by the updating the ordered allocation variables, (d_i). At most iterations this will be necessary for only a few indexes j. As for the weights, it is the updating of (α_j) what will determine how many entries of (p_j) must be updated. Thus, at most we will need to update p_j for $j \leq \max\{\alpha_1, \ldots, \alpha_J\}$ where J is the latest entry of (α_j) we were required to update.

Updating of m:

If the model dimension is random, our proposal is to update m, $(\tilde{x}_j)_{j>k_n}$, (p_j) and $(\alpha_j)_{j>k_n}$ as a block from the full conditional $\mathbb{p}[m, (\tilde{p}_j), (\tilde{x}_j)_{j>k_n}, (\alpha_j)_{j>k_n} \mid \cdots]$. Here we use "$\cdots$" to denote all random terms other than m, $(\tilde{x}_j)_{j>k_n}$, (p_j) and $(\alpha_j)_{j>k_n}$. Integrating over $(\tilde{x}_j)_{j>k_n}$, (p_j) and $(\alpha_j)_{j>k_n}$, we first sample m from

$$\mathbb{p}[m \mid \cdots] \propto \mathbb{E}\left[\prod_{j=1}^{k_n} p_{\alpha_j}^{n_j} \,\middle|\, m\right] \mathbb{1}_{\{m \geq k_n\}} \mathbb{p}[m], \tag{22}$$

where the expectation is taken with respect to the conditional distribution of (p_j) given m, and treating $(n_j)_{j \leq k_n}$ and $(\alpha_j)_{j \leq k_n}$ as constants. Later we sample $(\tilde{p}_j)$ and $(\alpha_j)_{j>k_n}$ from (17) as previously explained, and the $m - k_n$ empty component parameters, $\tilde{x}_j$, from the prior, cf. (11). In contrast to (15), the EPPF does not appear in (22), instead it is enough to compute an expectation of the weights. This is very convenient, being that when law of $(\tilde{p}_j)$ is not available, typically the EPPF is hard to compute as mentioned in Section 2.

3.3 Acceleration step

There is a very simple modification of the ordered allocation sampler that can greatly improve its performance. To motivate it, first note that the set of admissible moves, $\mathcal{D}_i$, of d_i is always

contained in $\{1, \ldots, k_{i-1}+1\}$, with $k_0 = 0$ and $k_i = \max\{d_1, \ldots, d_i\}$. Recalling that d_i indicates from which component of the mixture was y_i sampled, this means that while the latest data points will be able to reallocate to virtually all observed components, the first data points will rarely be reassigned to a different component. Furthermore, since component parameters and weights are labelled in the order in which they were discovered by (y_i), the initial order of data points can dictate how often there are label switches of components and thus affect the mixing properties of the sampler. To overcome this, it is enough to exploit the exchangeability of (y_i) and add a step, after updating (d_i), in which we randomly permute the data points obtaining $(y'_i) = (y_{\rho(i)})$, where ρ is a uniform permutation of $\{1, \ldots, n\}$. Accordingly, we modify (d_i) obtaining (d'_i) defined by $d'_i = j$ if and only if $d_{\rho(i)}$ equals the jth distinct value to appear in $(d_{\rho(i)})$. This way, the ordered allocation variables, (d'_i), that correspond to the permuted data set, (y'_i), preserve the induced clustering structure, and the least element order as dictated by the event $\mathcal{D}$ now holds for (D'_j) with $D'_j = \{i : d'_i = j\}$ (see Appendix D for an example). In accordance, for the sampler in Section 3.2 we will also need to change the values of $(\alpha_j)_{j \leq k_n}$ so to obtain $(\alpha'_j)_{j \leq k_n}$, where α'_j now indicates which weight in (p_j) is the jth one to be discovered by (y'_i). To do so we simply have to set $\alpha'_j = \alpha_l$ if and only if the jth distinct value to appear in $(d_{\rho(i)})$ equals l. After doing so we can move on with the updating of m (if it is random) and each of the observed component parameters, $\tilde{x}'_j$, and weights, $\tilde{p}'_j = p_{\alpha'_j}$, identically as before, although now they are labelled in order in which they were discovered by (y'_i).

For the simulation study we will present in the following section, this acceleration step was included in all implementations of the ordered allocation. Nonetheless, in Appendix E we present a few runs of the ordered allocation sampler without it to illustrate its effect.

4 Simulation study

In this section we present a simulation study to compare the mixing of the ordered allocation sampler against that of a marginal sampler and a conditional sampler. Following Kalli et al. (2011), three different data sets have been considered (histograms are displayed in Figure 1). The first data set is the galaxy data, consisting of the velocities of 82 distinct galaxies diverging away from our galaxy. The other two data sets are the leptokurtic and bimodal data sets first introduced in Green and Richardson (2001). The leptokurtic consists of 100 data points simulated from the mixture $0.67\,\mathsf{N}(0,1) + 0.33\,\mathsf{N}(0.3, 0.25^2)$. In the bimodal the 100 observations come from the mixture $0.5\,\mathsf{N}(-1, 0.5^2) + 0.5\,\mathsf{N}(1, 0.5^2)$. To each data set we fitted a mixture of Gaussian distributions with random location and scale parameters, i.e. $g(y \mid x) = \mathsf{N}(y \mid \mu, \sigma^2)$, and with five different mixing priors specifications. First we consider a mixture of finite mixtures (MFM, Miller and Harrison; 2018) specifically a mixing prior with random dimension, m, and symmetric Dirichlet weights, $(p_1, \ldots, p_m) \sim \mathsf{Dir}(\gamma, \ldots, \gamma)$, with $\gamma = 1$. As for m, we used the prior $\mathbb{p}[m] = \lambda(1-\lambda)_{m-1}/m!$ (Gnedin; 2010) with $\lambda = 0.1$. The remaining mixing priors we considered are a Dirichlet process (DP) with total mass parameter $\theta = 1$, a Pitman-Yor process (PY) with parameters $(\sigma, \theta) = (0.3, 0.7)$, a Geometric process (GP, Fuentes-García et al.; 2010) and an Exchangeable Stick-Breaking process (ESB, Gil–Leyva and Mena; 2021). Further specifications of the Geometric and the Exchangeable stick-breaking processes can be found in Appendix C. In all cases the distribution ν of the component parameters was fixed to $\nu(\mu, \sigma^2) = \mathsf{N}(\mu \mid \mu_0, \lambda_0^{-1}\sigma^2)\Gamma^{-1}(\sigma^2 \mid a_0, b_0)$ with hyperparameters $\mu_0 = n^{-1}\sum_{i=1}^n y_i$, $\lambda_0 = 1/100$ and $a_0 = b_0 = 0.5$.

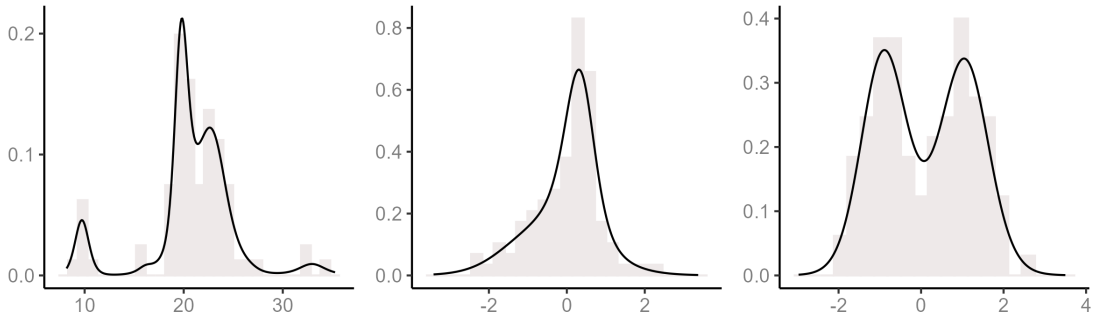

Figure 1: Histogram of the galaxy (left), leptokurtic (middle) and bimodal (right) datasets. The lines represent the estimated densities using the ESB mixing prior and the ordered allocation sampler in Section 3.2.

The marginal sampler we have implemented is Algorithm 8 in Neal (2000) for DP, PY and MFM. In particular for MFM, Algorithm 8 was adapted following Miller and Harrison (2018). No marginal samplers are available for GP and ESB as these priors lack a Pólya urn scheme representation. As for the conditional sampler, we implemented the (dependent) slice-efficient sampler, as described by Kalli et al. (2011), for all models but MFM. The ordered allocation sampler (OAS in short) was used to implement all considered mixing priors. In particular, we used the algorithm in Section 3.1 for MFM, DP and PY, as these priors enjoy a tractable law of the size-biased permuted weights. The algorithm of Section 3.2 was used for GP and ESB. In Appendix C we explain how to update the weights of the GP and ESB priors.

To monitor algorithmic performance we explored the convergence of the number of occupied components, k_n, and the deviance, D_v, of the estimated density (cf Green and Richardson; 2001). The deviance can be computed by $D_v = -2\sum_{i=1}^{n} \log \sum_{j=1}^{m} \frac{n_j}{n} g(y_i \mid x_j)$, where n_j is the number of data points associated to $g(y_i \mid x_j)$. More precisely, we considered the chains $(k_n^t)_{t=1}^T$ and $(D_v^t)_{t=1}^T$ attained from T iterations after the burn-in period. In each case we estimated the integrated autocorrelation time (IAT), $\tau = 1/2 + \sum_{l=1}^{\infty} \rho_l$, where ρ_l stands for l-lag autocorrelation of the monitored chain. As done by Kalli et al. (2011), τ was estimated through $\hat{\tau} = 1/2 + \sum_{l=1}^{C-1} \hat{\rho}_l$, where $\hat{\rho}_l$ is the estimated autocorrelation at lag l and $C = \min\{l : |\hat{\rho}_l| < 2/\sqrt{T}\}$. This is a very useful summary statistic for quantifying the convergence of an MCMC algorithm, smaller values of $\hat{\tau}$ corresponding to better performance. For each sampler and mixing prior, we considered 2×10^6 iterations after a burn-in period of 10^5 iterations. Table 1 reports estimates of the IAT for k_n and D_v with standard errors appearing in parenthesis, the latter computed following Section 3 of Sokal (1997).

We observe that, when applicable, Algorithm 8 outperforms the other samplers, and that the OAS has better mixing properties than the slice sampler. On average, the IAT corresponding to the OAS is roughly 1.8 times bigger than that of Algorithm 8, and the IAT of the slice sampler is approximately 4.7 times larger than that of the OAS. In general, it has been found that conditional algorithms perform worse than marginal samplers. This can be explained by the fact that in conditional algorithms mixing takes place in the space of all possible values of the (usual) allocation variables, $(c_i)_{i=1}^n$, given by $c_i = j$ if and only is $\theta_i = x_j$. Instead, in

		galaxy data			leptokurtic data		
		Marginal	OAS	Conditional	Marginal	OAS	Conditional
MFM	D_v	14.43(0.38)	26.17(0.93)	—	563.5(61.3)	855.5(94.8)	—
	k_n	33.55(0.92)	89.42(3.07)	—	337.7(26.3)	535.2(64.7)	—
DP	D_v	12.30(0.23)	23.76(0.57)	119.2(9.95)	22.42(0.54)	25.81(0.63)	120.5(10.3)
	k_n	13.68(0.25)	32.49(0.81)	190.2(16.3)	9.26(0.13)	18.99(0.41)	100.8(7.18)
PY	D_v	13.48(0.24)	21.59(0.52)	83.33(4.63)	53.85(1.45)	62.91(2.10)	322.9(37.0)
	k_n	12.43(0.23)	35.62(0.84)	115.7(6.23)	12.02(0.24)	20.96(0.56)	138.7(13.1)
GP	D_v	—	11.01(0.33)	40.76(2.50)	—	50.64(1.55)	158.7(7.86)
	k_n	—	61.67(1.89)	621.2(172.5)	—	45.34(1.21)	129.8(6.43)
ESB	D_v	—	24.29(0.68)	245.4(28.3)	—	61.29(1.97)	184.5(18.6)
	k_n	—	59.27(2.16)	632.4(88.9)	—	26.78(0.91)	120.1(10.7)

		bimodal data		
		Marginal	OAS	Conditional
MFM	D_v	122.9(8.04)	143.0(9.28)	—
	k_n	65.15(3.89)	109.4(7.99)	—
DP	D_v	7.84(0.16)	13.87(0.35)	35.61(1.68)
	k_n	6.30(0.07)	13.38(0.22)	52.00(2.18)
PY	D_v	39.91(1.33)	58.11(2.21)	257.1(22.8)
	k_n	6.24(0.14)	12.40(0.30)	58.10(3.35)
GP	D_v	—	57.45(1.76)	148.4(5.81)
	k_n	—	55.85(1.65)	146.0(5.94)
ESB	D_v	—	48.48(1.52)	76.51(3.48)
	k_n	—	19.93(0.58)	35.53(1.42)

Table 1: Results for the three datasets by model and sampler.

marginal algorithms the labels of the allocation variables are irrelevant, which means that the sampler searches is the space of partitions of $\{1, \dots, n\}$, generated by the ties among allocation variables (cf. Porteous et al.; 2006). Now, in the OAS, mixing occurs in the space of all possible values of the ordered allocation variables, (d_i), which unequivocally define an ordered partition of $\{1, \dots, n\}$, with blocks in the least element order. Since there exists a one to one correspondence between (unordered) partitions of $\{1, \dots, n\}$ and partitions, of the same set, ordered according to the least element, we find that marginal samplers and the OAS search in the exact same space. This explains the better mixing properties of the OAS when compared with the slice sampler. Still, Algorithm 8 has better performances compared with the OAS, which is mainly due to the restricted support of (d_i) in the OAS.

In Appendix E we extend this study for the DP model to further compare the distinct versions of the OAS in Sections 3.1 and 3.2, as well as the effect of the acceleration step in Section 3.3. There we also show the graph of the estimated weighted densities by component, so to illustrate in more details how the different samplers mix over component labels.

5 Discussion

The ordered allocation sampler exploits the conditional law of a species sampling sequence given the atoms and the weights in order of appearance. The idea of sorting the parameters by order of appearance is analogous to that of Chopin (2007) for devising sequential Monte Carlo algorithm for hidden Markov models. A key difference is that in our framework we retain exchangeability of the data, while in hidden Markov model the data possess a precise temporal order.

Mixture models with a random dimension have been long known for their appeal from a modelling perspective and for their optimal asymptotic properties (Rousseau and Mengersen;

2011; Shen et al.; 2013). However, posterior computation had remained somehow elusive until the advent of the marginal sampler by Miller and Harrison (2018). The ordered allocation sampler is a valid alternative, it is simple to implement, and more broadly applicable. The sampler has been illustrated for mixtures of finite mixtures, but it readily applies to symmetric Dirichlet distributed weights whose parameter γ can depend on the number of components. For example, we can use it to implement Dirichlet-multinomial mixing priors, or "sparse finite mixtures" as termed by Frühwirth-Schnatter and Malsiner-Walli (2019), where $\gamma = \theta/m$. As for mixtures with infinitely many components, the sampler completely avoids the truncation problem. In fact, it is practically identical for the case where m is a finite fixed number and the case where m is infinite. Other conditional samplers are designed for the case where $m < \infty$ is fixed, $m = \infty$, or m is random and there are clear distinctions between samplers that are designed for one case or another. To the best of our knowledge, the ordered allocation sampler is the first conditional sampler that treats in a unified manner the distinct assumptions on m.

As highlighted throughout the paper, the ordered allocation sampler enjoys nice properties in terms of applicability and mixing performance. Nonetheless, there are areas of improvements. In other conditional samplers allocation variables are updated independently of each other in a block, rather that one at a time. In big data settings this is a significant advantage over the ordered allocation sampler. Another drawback is that the number of occupied components, k_n, can not change as freely, from one iteration to the next one, as it does in other samplers. This explains the higher IAT of k_n when compared against the marginal method. The ordered allocation sampler may need additional modifications to address these issues, which is an interesting direction for future research.

References

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Polya urn schemes, *Ann. Statist.* **1**(2): 353 – 355.

Chopin, N. (2007). Inference and model choice for sequentially ordered hidden markov models, *JRSSB* **69**(2): 269–284.

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

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures, *J. Amer. Statist. Assoc.* **90**: 577–588.

Favaro, S., Lijoi, A., Nava, C., Nipoti, B., Prünster, I. and Teh, Y. W. (2016). On the stick-breaking representation for homogeneous NRMIs, *Bayesian Anal.* **11**: 697–724.

Favaro, S. and Teh, Y. W. (2013). MCMC for normalized random measure mixture models, *Statistical Science* **28**(3): 335 – 359.

Ferguson, T. (1973). A Bayesian analysis of some nonparametric problems, *Ann. Statist.* **1**(2): 209–230.

Frühwirth-Schnatter, S. and Malsiner-Walli (2019). From here to infinity: sparse finite versus dirichlet process mixtures in model-based clustering, *Advances in Data Analysis and Classification* **13**: 33–64.

Fuentes-García, R., Mena, R. H. and Walker, S. G. (2010). A new Bayesian nonparametric mixture model, *Communications in Statistics - Simulation and Computation* **39**(4): 669–682.

Gil–Leyva, M. F. and Mena, R. H. (2021). Stick-breaking processes with exchangeable length variables, *Journal of the American Statistical Association* p. in press.

Gnedin, A. (2010). A species sampling model with finitely many types, *Electronic Communications in Probability* **15**: 79–88.

Green, P. J. and Richardson, S. (2001). Modeling heterogeneity with and without the Dirichlet process, *Scand. J. Stat.* **28**: 355–375.

Ishwaran, H. and James, L. F. (2001). Gibbs sampling methods for stick-breaking priors, *J. Amer. Statist. Assoc.* **96**: 161–173.

Kallenberg, O. (2005). *Probabilistic Symmetries and Invariance Principles*, first edn, Springer.

Kalli, M., Griffin, J. E. and Walker, S. (2011). Slice sampling mixtures models, *Statist. Comput.* **21**: 93–105.

Lijoi, A., Prünster, I. and Rigon, T. (2020). The Pitman–Yor multinomial process for mixture modelling, *Biometrika* **107**(4): 891–906.

Miller, J. W. and Harrison, M. T. (2018). Mixture models with a prior on the number of components, *J. Amer. Statist. Assoc.* **113**(521): 340–356.

Neal, R. M. (2000). Markov Chain Sampling Methods for Dirichlet Process Mixture Models, *J. Comput. Graph. Statist.* **9**(2): 249–265.

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

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

Pitman, J. (1996a). Random discrete distributions invariant under size-biased permutation, *Adv. Appl. Probab.* **28**(2): 525–539.

Pitman, J. (1996b). Some developments of the Blackwell-MacQueen urn scheme, *in* T. F. et al. (ed.), *Statistics, Probability and Game Theory; Papers in honor of David Blackwell*, Vol. 30 of *Lecture Notes-Monograph Series*, Institute of Mathematical Statistics, Hayward, California, pp. 245–267.

Pitman, J. (2006). *Combinatorial Stochastic Processes*, Vol. 1875 of *École d'été de probabilités de Saint-Flour*, first edn, Springer-Verlag Berlin Heidelberg, New York.

Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator, *Ann. Probab.* **25**(2): 855–900.

Porteous, I., Ihler, A., Smyth, P. and Welling, M. (2006). Gibbs sampling for (coupled) infinite mixture models in the stick breaking representation, *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence (UAI2006)*, pp. 385–392.

Regazzini, E., Lijoi, A. and Prünster, I. (2003). Distributional results for means of normalized random measures with independent increments, *Ann. Statist.* **31**(2): 560–585.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components, *J. R. Stat. Soc. Ser. B* **59**: 731–792.

Rodríguez, A. and Dunson, D. B. (2011). Nonparametric Bayesian models through probit stick-breaking processes, *Bayesian Anal.* **6**(1): 145–178.

Rousseau, J. and Mengersen, K. (2011). Asymptotic behaviour of the posterior distribution in overfitted mixture models, *J. R. Stat. Soc. Ser. B* **73**(5): 689–710.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors, *Stat. Sin.* **4**: 639–650.

Shen, W., Tokdar, S. T. and Ghosal, S. (2013). Adaptive Bayesian multivariate density estimation with Dirichlet mixtures, *Biometrika* **100**: 623–640.

Sokal, A. (1997). Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms, *in* C. DeWitt-Morette, P. Cartier and A. Folacci (eds), *Functional Integration: Basics and Applications*, Springer US, Boston, MA, pp. 131–192.

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

Zanella, G. (2020). Informed proposals for local MCMC in discrete spaces, *J. Amer. Statist. Assoc.* **115**(530): 852–865.

Appendix

A Proof of Theorem 1

Lemma A.1. *Let m be a random variable taking values in $\{1, 2, \ldots\} \cup \{\infty\}$ and let $(\tilde{p}_j) = (\tilde{p}_1, \ldots, \tilde{p}_m)$ be a sequence in $[0, 1]$ with $\sum_{j=1}^{m} \tilde{p}_j = 1$. Let $\pi : \bigcup_{k\in\mathbb{N}} \mathbb{N}^k \to [0, 1]$ be defined by*

$$\pi(n_1, \ldots, n_k) = \mathbb{E}\Big[\prod_{j=1}^{k} \tilde{p}_j^{n_j-1} \prod_{j=1}^{k-1} \Big(1 - \sum_{l=1}^{j} \tilde{p}_l\Big)\Big].$$

Then $(\tilde{p}_j)$ is invariant under size-biased permutations if and only if π is a symmetric function of $(n_1, \ldots, n_k)$.

The proof of Lemma A.1 can be found in Pitman (1995, 1996a). Actually, Pitman derived it more in general for a sequence $(\tilde{p}_j)_{j=1}^{\infty}$ taking values in the infinite dimensional simplex $\Delta_\infty = \Big\{(w_j)_{j=1}^{\infty} : w_j \geq 0, \sum_{j=1}^{\infty} w_j = 1\Big\}$. The statement in Lemma A.1 easily follows by transforming $(\tilde{p}_1, \ldots, \tilde{p}_m)$ into a sequence in Δ_∞ by appending zeros, i.e. $\tilde{p}_j = 0$ for $j > m$. For simplicity we will first take for granted Lemma A.1, later in Remark A.1 we explain how to derive a self-contained proof.

Proof of Theorem 1:

(Sufficiency): Assume (θ_i) is a species sampling sequence driven by the species sampling model $P = \sum_{j=1}^{m} p_j \delta_{x_j}$. By de Finetti's theorem,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \delta_{\theta_i} = \sum_{j=1}^{m} p_j \delta_{x_j},$$

almost surely. As $\sum_{j=1}^{m} p_j = 1$ and $p_j > 0$, we get that outside a $\mathbb{P}$-null event, $\theta_i \in \{x_1, \ldots, x_m\}$ for every $i \geq 1$, and for each $j \leq m$ there exist $i \geq 1$ such that $\theta_i = x_j$. This together with the diffuseness of ν, yield that (θ_i) exhibits exactly m distinct values, $x_1, \ldots, x_m$, almost surely. This means that we can define $(\alpha_j) = (\alpha_1, \ldots, \alpha_m)$ given by $\alpha_l = j$ if and only if $\theta_{M_l} = x_j$, recalling that $M_1 = 1$ and for $2 \leq l \leq m$, $M_l = \min\{i > M_{l-1} : \theta_i \notin \{\theta_{M_1}, \ldots, \theta_{M_{l-1}}\}\}$. This way, $\tilde{x}_j = x_{\alpha_j}$ is the jth distinct value of (θ_i) in order of appearance. Next we prove that (α_j) satisfies equation (3) in the main document. To this aim, note that P is $\{(p_j), (x_j)\}$-measurable and vice versa. Moreover $(\theta_{M_1}, \ldots, \theta_{M_l})$ is $\{(\alpha_1, \ldots, \alpha_l), (x_j)\}$-measurable and $(\alpha_1, \ldots, \alpha_l)$ is $\{(\theta_{M_1}, \ldots, \theta_{M_l}), (x_j)\}$-measurable. As (θ_i) is conditionally iid from P, this implies

$$\mathbb{P}[\alpha_1 = j \mid (p_j), (x_j)] = \mathbb{P}[\theta_1 = x_j \mid P] = p_j$$

and for $2 \leq l \leq m$,

$$\begin{aligned}\mathbb{P}[\alpha_l = j \mid (p_j), (x_j), \alpha_1, \ldots, \alpha_{l-1}] &= \mathbb{P}[\theta_{M_l} = x_j \mid P, \theta_{M_1}, \ldots, \theta_{M_{l-1}}] \\ &\propto p_j \mathbb{1}_{\{x_j \notin \{\theta_{M_1}, \ldots, \theta_{M_{l-1}}\}\}}\end{aligned}$$

This is

$$\mathbb{P}[\alpha_l = j \mid m, (p_j), (x_j), \alpha_1, \ldots, \alpha_{l-1}] = \frac{p_j}{1 - \sum_{i=1}^{l-1} p_{\alpha_i}} \mathbb{1}_{\{j \notin \{\alpha_1, \ldots, \alpha_{l-1}\}\}}.$$

Hence, given (p_j), (α_j) satisfies (3). Moreover, as (p_j) is independent of (x_j), we also get that (α_j) is independent of (x_j), which are iid from ν. This yields that $(\tilde{x}_j) = (x_{\alpha_j})$ are also iid from ν, so i is proved.

Using de Finetti's theorem once more, we get

$$\lim_{n\to\infty} \sum_{j=1}^{k_n} \frac{|\{i \le n : \theta_i = \tilde{x}_j\}|}{n} \delta_{\tilde{x}_j} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \delta_{\theta_i} = \sum_{j=1}^{m} p_j \delta_{x_j} = \sum_{j=1}^{m} p_{\alpha_j} \delta_{x_{\alpha_j}},$$

where k_n is the number of distinct values in $\{\theta_1, \ldots, \theta_n\}$. Since the directing random measure of an exchangeable sequence is unique almost surely, this assures $k_n \to m$, and for $j \le m$, the long run proportion of indexes i such that $\theta_i = \tilde{x}_j = x_{\alpha_j}$ is

$$\tilde{p}_j = \lim_{n\to\infty} \frac{|\{i \le n : \theta_i = \tilde{x}_j\}|}{n} = p_{\alpha_j},$$

almost surely. As (3) holds for (α_j), $(\tilde{p}_j) = (p_{\alpha_j})$ is a size-biased permutation of (p_j), which yields *ii*.

As for *iii*, first note that by definition, $\tilde{x}_1, \ldots, \tilde{x}_{k_n}$ are the k_n distinct values in order of appearance in $\{\theta_1, \ldots, \theta_n\}$, for every $n \ge 1$, in particular $\theta_1 = \tilde{x}_1$. Now, as (θ_i) is conditionally iid from $P = \sum_{j=1}^{m} p_j \delta_{x_j} = \sum_{j=1}^{m} \tilde{p}_j \delta_{\tilde{x}_j}$, we get that for each $n \ge 1$ and every $j \le k_n$,

$$\mathbb{P}[\theta_{n+1} = \tilde{x}_j \mid (\tilde{p}_j), (\tilde{x}_j), \theta_1, \ldots, \theta_n] = \tilde{p}_j.$$

Thus,

$$\mathbb{P}[\theta_{n+1} \notin \{\tilde{x}_1, \ldots, \tilde{x}_{k_n}\} \mid (\tilde{p}_j), (\tilde{x}_j), \theta_1, \ldots, \theta_n] = 1 - \sum_{j=1}^{k_n} \tilde{p}_j.$$

By definition, under the event $\theta_{n+1} \notin \{\tilde{x}_1, \ldots, \tilde{x}_{k_n}\}$ we must have $\theta_{n+1} = \tilde{x}_{k_n+1}$, i.e.

$$\mathbb{P}[\theta_{n+1} \in \cdot \mid (\tilde{x}_j), (\tilde{p}_j), \theta_1, \ldots, \theta_n] = \sum_{j=1}^{k_n} \tilde{p}_j \delta_{\tilde{x}_j} + \left(1 - \sum_{j=1}^{k_n} \tilde{p}_j\right) \delta_{\tilde{x}_{k_n+1}}.$$

As for *iv*, first note that $(\tilde{p}_j)$ is $\{(p_j), (\alpha_j)\}$-measurable and (p_j) is $\{(\tilde{p}_j), (\alpha_j)\}$-measurable. The last assertion relies on $p_j = \tilde{p}_{\alpha_j^{-1}}$, where (α_j^{-1}) is the inverse permutation of (α_j). From the proof of i and the hypothesis, we have that (α_j), (p_j) and m are independent of (x_j). Thus, for every $B \in \mathcal{B}(\mathbb{X})$,

$$\mathbb{P}[\tilde{x}_j \in B \mid m, (\tilde{p}_j), (\alpha_j)] = \mathbb{P}[\tilde{x}_j \in B \mid m, (p_j), (\alpha_j)] = \mathbb{P}[x_{\alpha_j} \in B \mid m, (p_j), (\alpha_j)] = \nu(B),$$

which proves *iv*.

(Necessity): Assume *i*–*iv* hold. We first prove that (θ_i) is exchangeable. Fix $n \ge 1$ and define the random partition, Π_n of $[n] = \{1, \ldots, n\}$ generated by the random equivalence relation $i \sim j$

if and only of $\theta_i = \theta_j$. In other words $\Pi_n = \{D_1, \dots, D_{k_n}\}$ where $D_j = \{i \leq n : \theta_i = \tilde{x}_j\}$. Using *iii*, a simple counting argument implies

$$\mathbb{P}[\Pi_n = \{A_1, \dots, A_k\} \mid (\tilde{p}_j), (\tilde{x}_j)] = \prod_{j=1}^{k} \tilde{p}_j^{n_j - 1} \prod_{j=1}^{k-1} \Big(1 - \sum_{l=1}^{j} \tilde{p}_j\Big), \tag{A1}$$

for every partition $\{A_1, \dots, A_k\}$ of $[n]$, and where $n_j = |A_j|$. Taking expectations in (A1),

$$\mathbb{P}[\Pi_n = \{A_1, \dots, A_k\}] = \mathbb{E}\Big[\prod_{j=1}^{k} \tilde{p}_j^{n_j - 1} \prod_{j=1}^{k-1} \Big(1 - \sum_{l=1}^{j} \tilde{p}_j\Big)\Big].$$

By *ii* and Lemma A.1, the function

$$(n_1, \dots, n_k) \mapsto \pi(n_1, \dots, n_k) = \mathbb{E}\Big[\prod_{j=1}^{k} \tilde{p}_j^{n_j - 1} \prod_{j=1}^{k-1} \Big(1 - \sum_{l=1}^{j} \tilde{p}_j\Big)\Big]$$

is symmetric. This shows $\mathbb{P}[\Pi_n = \{A_1, \dots, A_k\}] = \pi(n_1, \dots, n_k)$, at most depends on the number of blocks k of $\{A_1, \dots, A_k\}$ and the frequencies, $n_1, \dots, n_k$, of each block, through a symmetric function. In other words, Π_n is exchangeable, in the sense that for every permutation, ρ, of $[n]$, Π_n is equal in distribution to $\rho(\Pi_n)$, where

$$\rho(\Pi_n) = \{\rho(D_j) : D_j \in \Pi_n\} \quad \text{and} \quad \rho(D_j) = \{\rho(i) : i \in D_j\}.$$

Now, fix $B_1, \dots, B_n \in \mathcal{B}(\mathbb{X})$ and note

$$\mathbb{P}[\theta_{\rho(1)} \in B_1, \dots, \theta_{\rho(n)} \in B_n \mid \Pi_n] = \mathbb{P}[\tilde{x}_j \in B_l, \forall\, l \in \rho(D_j), j \leq k_n \mid \Pi_n] = \prod_{j=1}^{k_n} \nu\Big(\bigcap_{i \in \rho(D_j)} B_i\Big).$$

The last equality follows from the fact that $(\tilde{x}_j)$ are iid from ν, and $\tilde{x}_j$ is independent of m and $(\tilde{p}_j)$, which together with (A1) imply $\tilde{x}_j$ is independent of Π_n. By taking expectations in the last equation we find,

$$\mathbb{P}[\theta_{\rho(1)} \in B_1, \dots, \theta_{\rho(n)} \in B_n] = \mathbb{E}\Big[\prod_{j=1}^{k_n} \nu\Big(\bigcap_{i \in \rho(D_j)} B_i\Big)\Big].$$

As Π_n is exchangeable,

$$\mathbb{E}\Big[\prod_{j=1}^{k_n} \nu\Big(\bigcap_{i \in \rho(D_j)} B_i\Big)\Big] = \mathbb{E}\Big[\prod_{j=1}^{k_n} \nu\Big(\bigcap_{i \in D_j} B_i\Big)\Big],$$

hence

$$\mathbb{P}[\theta_{\rho(1)} \in B_1, \dots, \theta_{\rho(n)} \in B_n] = \mathbb{P}[\theta_1 \in B_1, \dots, \theta_n \in B_n],$$

which proves (θ_i) is exchangeable. Finally, by de Finetti's theorem we know that directing random measure of (θ_i) is given by

$$P = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \delta_{\theta_i} = \lim_{n \to \infty} \frac{|\{i \leq n : \theta_i = \tilde{x}_j\}|}{n} \sum_{j=1}^{k_n} \delta_{\tilde{x}_j},$$

and by i and ii we conclude

$$P = \sum_{j=1}^{m} \tilde{p}_j \delta_{\tilde{x}_j} = \sum_{j=1}^{m} p_{\alpha_j} \delta_{x_{\alpha_j}} = \sum_{j=1}^{m} p_j \delta_{x_j}.$$

□

Remark A.1. The sufficiency of Lemma A.1, which we require to prove the necessity of Theorem 1, can be easily derived using the sufficiency of Theorem 1, thus provide a self-contained proof of Theorem 1. Namely, in the context of Lemma A.1 let $(\tilde{p}_j)$ be invariant under size-biased permutations, and let $(p'_j) = (p'_1, \ldots, p'_{m'})$ be any sequence of weights whose size-biased permutation has the law of $(\tilde{p}_j)$. Then we can construct a species sampling model $P' = \sum_{j=1}^{m'} p'_j \delta_{x'_j}$ over a Borel space $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$ and a species sampling sequence (θ'_i) driven by P'. By the sufficiency of Theorem 1, we get that for every $n \geq 1$,

$$\mathbb{P}[\theta'_{n+1} \in \cdot \mid (\tilde{p}'_j), (\tilde{x}'_j), \theta'_1, \ldots, \theta'_n] = \sum_{j=1}^{k'_n} \tilde{p}'_j \delta_{\tilde{x}'_j} + \Big(1 - \sum_{j=1}^{k_n} \tilde{p}'_j\Big) \delta_{\tilde{x}'_{k'_n+1}}, \tag{A2}$$

where $(\tilde{x}'_j)$ are the distinct values that (θ'_i) exhibits in order of appearance, and $(\tilde{p}'_j)$ denotes the size-biased permutation of (p'_j). Now, let Π'_n be the random partition of $[n]$ generated by the random equivalence relation $i \sim j$ if and only if $\theta'_i = \theta'_j$. Then, by construction Π'_n is exchangeable, and a simple counting argument, using (A2), yields

$$\mathbb{P}[\Pi'_n = \{A_1, \ldots, A_k\}] = \mathbb{E}\Big[\prod_{j=1}^{k} \Big(\tilde{p}'_j\Big)^{n_j - 1} \prod_{j=1}^{k-1} \Big(1 - \sum_{l=1}^{j} \tilde{p}'_j\Big)\Big],$$

where $n_j = |A_j|$. Since $(\tilde{p}'_j)$ is equal in distribution to $(\tilde{p}_j)$ and Π'_n is exchangeable, we conclude

$$\pi(n_1, \ldots, n_k) = \mathbb{E}\Big[\prod_{j=1}^{k} \tilde{p}_j^{n_j - 1} \prod_{j=1}^{k-1} \Big(1 - \sum_{l=1}^{j} \tilde{p}_j\Big)\Big] = \mathbb{E}\Big[\prod_{j=1}^{k} \Big(\tilde{p}'_j\Big)^{n_j - 1} \prod_{j=1}^{k-1} \Big(1 - \sum_{l=1}^{j} \tilde{p}'_j\Big)\Big]$$

is a symmetric function of $(n_1, \ldots, n_k)$.

B Mixtures of finite mixtures with Gnedin (2010) prior on m

In this section we illustrate how to update the model dimension m by sampling from (15). We consider a mixture model with symmetric Dirichlet weights $(p_1, \ldots, p_m) \sim \mathsf{Dir}(1, \ldots, 1)$, and random m with prior distribution

$$\mathbb{p}[m] = \frac{\lambda(1-\lambda)_{m-1}}{m!},$$

where $\lambda \in (0, 1)$ is a known constant. As mentioned in Section 2, the size-biased permuted weights $(\tilde{p}_j)$ admit stick-breaking representation $\tilde{p}_1 = v_1$, and $\tilde{p}_j = v_j \prod_{i=1}^{j-1}(1 - v_i)$ for independent random variables, $v_j \sim \mathsf{Be}(1 - \sigma, \theta + j\sigma)$ where $\sigma = -1$ and $\theta = m$. Thus, the ordered allocation sampler as derived in Section 3.1 can be used to implement this model.

First note that using (4) and the stick-breaking decomposition of $(\tilde{p}_j)$, we can compute the conditional EPPF given m:

$$\pi(n_1, \ldots, n_{k_n} \mid m) = \frac{\prod_{j=1}^{k_n-1}(\theta + j\sigma)}{(\theta+1)_{n-1}} \prod_{j=1}^{k_n} (1-\sigma)_{n_j-1} = \frac{(m-k_n+1)_{k_n-1}}{(m+1)_{n-1}} \prod_{j=1}^{k_n} n_j!$$

where $n = \sum_{j=1}^{k_n} n_j$ and $(z)_r = \prod_{i=0}^{r-1}(z+i)$, using the convention that the empty product equals one. Hence, (15) simplifies to

$$\mathbb{p}[\, m \mid \cdots] \propto \frac{(m-k_n+1)_{k_n-1}}{(m+1)_{n-1}} \times \frac{\lambda(1-\lambda)_{m-1}}{m!} \mathbb{1}_{\{k_n \le m\}}.$$

Following Gnedin (2010), we obtain

$$\sum_{m=k_n}^{\infty} \frac{(m-k_n+1)_{k_n-1}}{(m+1)_{n-1}} \frac{\lambda(1-\lambda)_{m-1}}{m!} = \frac{(k_n-1)!(1-\lambda)_{k_n-1}(\lambda)_{n-k_n}}{(n-1)!(1+\lambda)_{n-1}}.$$

Thus, we can explicitly compute

$$\mathbb{p}[\, m \mid \cdots] = \frac{\lambda(1-\lambda)_{m-1}(m-k_n+1)_{k_n-1}}{(m+n-1)!} \times \frac{(n-1)!(1+\lambda)_{n-1}}{(k_n-1)!(1-\lambda)_{k_n-1}(\lambda)_{n-k_n}} \mathbb{1}_{\{k_n \le m\}}.$$

In particular, using notation $q_r = \mathbb{p}[\, m = r \mid \cdots]$

$$q_{k_n} = \frac{(\lambda + n - k_n)_{k_n}}{(n)_{k_n}},$$

and recursively for $r \ge k_n$,

$$q_{r+1} = q_r \frac{r(r-\lambda)}{(r-k_n+1)(r+n)}.$$

Thus, to update m, sample $u \sim \mathsf{Unif}(0,1)$, and set $m = r$ when $\sum_{l=k_n}^{r-1} q_l < u \le \sum_{l=k_n}^{r} q_l$.

C Geometric and exchangeable stick-breaking processes

To illustrate the ordered allocation sampler derived in Section 3.2 we chose two species sampling mixing priors for which the law of $(\tilde{p}_j)$ is not available. These are the geometric process and the exchangeable stick-breaking process. The geometric process (Fuentes-García et al.; 2010) is a species sampling model $P = \sum_{j=1}^{\infty} p_j \delta_{x_j}$ with decreasingly ordered weights, (p_j), given by $p_j = v(1-v)^{j-1}$ where v is a random variable taking values in $(0,1)$. The exchangeable stick-breaking process (Gil–Leyva and Mena; 2021) instead has weights $p_j = v_j \prod_{l=1}^{j-1}(1-v_l)$, where $(v_j) = (v_j)_{j=1}^{\infty}$ is an exchangeable sequence with values in $(0,1)$. Here we consider (v_j) to be a species sampling sequence driven by a Dirichlet process P' over $([0,1], \mathcal{B}([0,1]))$, with total mass parameter θ' and base measure $\nu' = \mathsf{Be}(a,b)$. Next we refer to $P = \sum_{j=1}^{\infty} p_j \delta_{x_j}$ as Dirichlet driven exchangeable stick-breaking process.

To fully specialize the ordered allocation sampler in Section 3.2 for this two mixing priors it is enough to explain how to update (p_j) via sampling (v_j) from

$$\mathbb{p}[\,(v_j) \mid \cdots] \propto \prod_{j=1}^{\overline{\alpha}} v_j^{r_j}(1-v_j)^{R_j}\,\mathbb{p}[(v_j)], \tag{C3}$$

where $r_j = \sum_{l=1}^{k_n} n_l \mathbb{1}_{\{\alpha_l = j\}}$, $R_j = \sum_{l>j} r_l$, $\overline{\alpha} = \max\{\alpha_1, \ldots, \alpha_{k_n}\}$, and "$\cdots$" refers to all the random variables involved excluding $(\alpha_j)_{j>k_n}$ and (p_j). Note that by excluding (p_j) we are also excluding (v_j) because these two sequences characterize each other. It is worth noting that the following description can be readily adapted to the updating of (v_j) in the slice-efficient sampler.

For the geometric process we have that $v_j = v$ for every $j \geq 1$, hence it suffices to update v from

$$\mathbb{p}[\,v \mid \cdots] \propto v^n (1-v)^{\sum_{i=1}^n c_i - n} \times \mathbb{p}[v],$$

where $c_i = \alpha_{d_i}$ for each $i \leq n$. In particular if $v \sim \mathsf{Be}(a,b)$ a priori, then we update $v \sim \mathsf{Be}(a+n, b+\sum_{i=1}^n c_i - n)$.

Now, for Dirichlet driven exchangeable stick-breaking processes, the updating of (v_j) is more delicate due to the non-trivial dependence among elements in (v_j). We will first focus on updating $(v_j)_{j\leq\overline{\alpha}}$. To this aim note that since (v_j) is a species sampling sequence driven by a Dirichlet process with total mass parameter θ' and base measure $\nu' = \mathsf{Be}(a,b)$, we can compute

$$\mathbb{p}[(v_j)_{j\leq\overline{\alpha}}] = \frac{(\theta')^{k_{\overline{\alpha}}}}{(\theta')_n} \prod_{l=1}^{k_{\overline{\alpha}}} (m_l - 1)!\mathsf{Be}(v_l^* \mid a,b),$$

where $(v_l^*) = (v_1^*, \ldots, v_{k_{\overline{\alpha}}}^*)$ are the distinct values that $(v_j)_{j\leq\overline{\alpha}}$ exhibits, $m_l = |E_l|$ and $E_l = \{j \leq \overline{\alpha} : v_j = v_l^*\} = \{j \leq \overline{\alpha} : e_j = l\}$, with $e_j = l$ if and only if $v_j = v_l^*$ (cf. Pitman; 1996b; Neal; 2000). Thus (C3) yields

$$\mathbb{p}[\,(v_l^*), (e_j) \mid \cdots] \propto \frac{(\theta')^{k_{\overline{\alpha}}}}{(\theta')_n} \prod_{l=1}^{k_{\overline{\alpha}}} (m_l - 1)!(v_l^*)^{\sum_{j\in E_l} r_j}(1-v_l^*)^{\sum_{j\in E_l} R_j}\mathsf{Be}(v_l^* \mid a,b).$$

Now to update (v_j) we can first sample (v_l^*) from

$$\mathbb{p}[\,(v_l^*) \mid (e_j), \cdots] \propto \prod_{l=1}^{k_{\overline{\alpha}}} \mathsf{Be}\left(v_l^* \,\middle|\, a + \sum_{j\in E_l} r_j, b + \sum_{j\in E_l} R_j\right),$$

which is a product of independent Beta distributions. Afterwards, for each $j \leq \overline{\alpha}$, we can update which value does v_j take among the ones observed in the rest of the v_j's or if it takes a new unobserved value. Say that $(v_l^*)_{-j} = (v_1^*, \ldots, v_{k_{-j}}^*)$ are the distinct values in $(v_i : i \leq \overline{\alpha}, i \neq j)$, in no particular order, and assume without loss of generality that $e_i = l$ if and only if $v_i = v_l^*$ for each $i \neq l$. Then it is enough to sample from

$$\mathbb{p}[e_j = e \mid (e_i)_{i\neq j}, (v_l^*)_{-j}, \cdots] \propto \begin{cases} m_e (v_e^*)^{r_j}(1-v_e^*)^{R_j} & \text{if } e \in \{1, \ldots, k_{-j}\} \\ \theta' \int v^{r_j}(1-v)^{R_j}\mathsf{Be}(dv \mid a,b) & \text{if } e = k_{-j}+1 \\ 0 & \text{otherwise,} \end{cases}$$

where $m_e = |\{i \neq j : e_i = e\}|$ and

$$\int v^{r_j}(1-v)^{R_j}\,\mathsf{Be}(dv \mid a,b) = \frac{\Gamma(r_j+a)\Gamma(R_j+b)}{\Gamma(r_j+R_j+a+b)}.$$

If the updated value $e_j \in \{1,\dots,k_{-j}\}$ we simply set $v_j = v^*_{e_j}$ otherwise if $e_j = k_{-j}+1$ we sample $v_j \sim \mathsf{Be}(a+r_j, b+R_j)$. Once we have updated $(v_j)_{j\leq\overline{\alpha}}$, we can update v_j for $j > \overline{\alpha}$ by sampling sequentially from $\mathbb{p}[\,v_j \mid (v_i)_{i<j}, \cdots]$, which happens to coincide with the prior prediction rule $\mathbb{p}[\,v_j \mid (v_i)_{i<j}\,]$. As (v_j) is a species sampling sequence driven by a Dirichlet process, P', with total mass parameter θ' and base measure $\nu' = \mathsf{Be}(a,b)$, it is well known that

$$\mathbb{P}[v_j \in \cdot \mid (v_i)_{i<j}] = \sum_{j=1}^{k_{j-1}} \frac{m_l}{\theta'+j-1}\delta_{v^*_l} + \frac{\theta'}{\theta'+j-1}\nu'$$

where $v^*_1,\dots,v^*_{k_{j-1}}$ are the distinct values in $(v_i)_{i<j}$, and $m_l = |\{i<j : v_i = v^*_l\}|$ (cf. Pitman; 1996b). Thus updating v_j for $j > \overline{\alpha}$ is easy, and it will be required only for a few $j > \overline{\alpha}$.

In general, this way of updating (v_j) for Dirichlet driven exchangeable stick-breaking processes is actually an adaptation of Algorithm 2 in Neal (2000), however, other marginal methods such as Algorithm 8 can also be exploited. In fact, by taking into account the underlying Dirichlet process, P', of (v_j), even a version of the slice sampler or the ordered allocation sampler could have been used. To conclude, we mention that for the simulation study in Section 4 of the main document we fixed the hyperparameters $\theta' = 1$, and $a = b = 1$ for both geometric and Dirichlet driven exchangeable stick-breaking models.

D Ordered allocation variables

Here we discuss the set $\mathcal{D}_i$ of admissible moves for the updating of the ordered allocation variable d_i. Some general rules for determining $\mathcal{D}_i$ can be envisioned: (i) if d_i is different from any other d_j, that is $D_{d_i} = \{i\}$, then d_i cannot change, unless $d_i = k_n$; and (ii) $\mathcal{D}_i \subset \{1,\dots,k_{i-1}+1\}$, so for larger i, there are more possible admissible moves, in particular, $d_1 = 1$ cannot change. As for illustration, let $n = 5$ and say that before updating (d_i), $d_1,\dots,d_5$ are such that the blocks of the partition in the least element order are $D_1 = \{1,3\}$, $D_2 = \{2,4\}$ and $D_3 = \{5\}$. Clearly $d_1 = 1$ cannot change. As for d_2, the admissible moves are $\mathcal{D}_2 = \{1,2\}$ thus d_2 will be sampled from

$$\mathbb{p}(d_2 = 1 \mid \text{rest}) \propto \tilde{p}_1 g(y_2 \mid \tilde{x}_1), \quad \mathbb{p}(d_2 = 2 \mid \text{rest}) \propto \tilde{p}_2 g(y_2 \mid \tilde{x}_2).$$

Say that we sample $d_2 = 1$, so that now $D_1 = \{1,2,3\}$, $D_2 = \{4\}$, $D_3 = \{5\}$. Since the blocks must be in least element order, the admissible moves for d_3 are $\mathcal{D}_3 = \{1,2\}$, hence d_3 will be sampled from

$$\mathbb{p}(d_3 = 1 \mid \text{rest}) \propto \tilde{p}_1 g(y_3 \mid \tilde{x}_1), \quad \mathbb{p}(d_3 = 2 \mid \text{rest}) \propto \tilde{p}_2 g(y_3 \mid \tilde{x}_2).$$

Assume we sample $d_3 = 1$ so now $D_1 = \{1,2,3\}$, $D_2 = \{4\}$, and $D_3 = \{5\}$. Given that D_2 can not be an empty set, under the current configuration, the only admissible move for d_4 is $\mathcal{D}_4 = \{4\}$, i.e. d_4 cannot change. Finally, the admissible moves for d_5 are $\mathcal{D}_5 = \{1,2,3\}$, and we will sample d_5 from

$$\mathbb{p}(d_5 = 1 \mid \text{rest}) \propto \tilde{p}_1 g(y_5 \mid \tilde{x}_1), \quad \mathbb{p}(d_5 = 2 \mid \text{rest}) \propto \tilde{p}_2 g(y_5 \mid \tilde{x}_2),$$

$$\mathbb{p}(d_5 = 3 \mid \text{rest}) \propto (1 - \tilde{p}_1 - \tilde{p}_2) g(y_5 \mid \tilde{x}_3).$$

Finally, assuming that we sample $d_5 = 2$, the initial configuration $D_1 = \{1, 3\}$, $D_2 = \{2, 4\}$ and $D_3 = \{5\}$, is updated to $D_1 = \{1, 2, 3\}$ and $D_2 = \{4, 5\}$.

In Section 3.3 we discuss an acceleration step that consists in randomly permuting the data points at each iteration of the sampler. Next we provide an example on how to modify the ordered allocation variables so to preserve the induced clustering structure, as well as the least element order of the partition induced by the modified variables. Let us consider again, as starting values of (d_i) the ones corresponding to the partition $D_1 = \{1, 3\}$, $D_2 = \{2, 4\}$ and $D_3 = \{5\}$, so

$$(d_i) = (1, 2, 1, 2, 3).$$

Also consider the permutation $\rho = (2, 1, 3, 5, 4)$, i.e. $\rho(1) = 2, \rho(2) = 1, \ldots \rho(5) = 4$, so that the permuted data set is $(y'_i) = (y_{\rho(i)}) = (y_2, y_1, y_3, y_5, y_4)$. The ordered allocation variables, (d_i), induce the following clustering of data points:

$$\{y_1, y_3\} = \{y'_2, y'_3\}, \quad \{y_2, y_4\} = \{y'_1, y'_5\}, \quad \{y_5\} = \{y'_4\}.$$

Thus, the original partition $(\{1, 3\}, \{2, 4\}, \{5\})$ becomes $(D'_1, D'_2, D'_3) = (\{1, 5\}, \{2, 3\}, \{4\})$ with respect to the new labeling of the observations. Let us check that the ordered allocation variables, $(d'_i) = (1, 2, 2, 3, 1)$, that correspond to (D'_j), can be obtained as explained in Section 3.3, that is $d'_i = j$ if and only if $d_{\rho(i)}$ equals the jth distinct value to appear in $(d_{\rho(i)})$. We have

$$(d_{\rho(i)}) = (d_2, d_1, d_3, d_5, d_4) = (2, 1, 1, 3, 2)$$

so that the distinct values of $(d_{\rho(i)})$ in order of appearance are $2, 1, 3$. Then,

$$\begin{aligned}
&d_{\rho(1)} = d_2 = 2 \text{ is the first distinct value to appear in } (d_{\rho(i)}) \text{ hence } d'_1 = 1,\\
&d_{\rho(2)} = d_1 = 1 \text{ is the second distinct value to appear in } (d_{\rho(i)}) \text{ hence } d'_2 = 2,\\
&d_{\rho(3)} = d_3 = 1 \text{ is the second distinct value to appear in } (d_{\rho(i)}) \text{ hence } d'_3 = 2,\\
&d_{\rho(4)} = d_5 = 3 \text{ is the third distinct value to appear in } (d_{\rho(i)}) \text{ hence } d'_4 = 3,\\
&d_{\rho(5)} = d_4 = 2 \text{ is the first distinct value to appear in } (d_{\rho(i)}) \text{ hence } d'_5 = 1.
\end{aligned}$$

We conclude that

$$(d'_i) = (1, 2, 2, 3, 1),$$

which in fact yields the partition in least element order $D'_1 = \{1, 5\}, D'_2 = \{2, 3\}, D'_3 = \{4\}$.

E Extended simulation study for the DP model

In this section we provide further illustrations of the two versions of the ordered allocation sampler derived in Sections 3.1 and 3.2 of the main paper, and of the importance of the acceleration step in Section 3.3. We consider the Dirichlet process mixing prior and we repeat the simulation study of Section 4 implementing, together with Algorithm 8 in Neal (2000) (Marginal), the dependent slice-efficient sampler by Kalli et al. (2011) (Conditional) and the sampler of Section 3.1 with the acceleration step (OAS1), the sampler of Section 3.2 with the acceleration step (OAS2) and the sampler of Section 3.1 without the acceleration step (OAS1*). Table E1 reports the

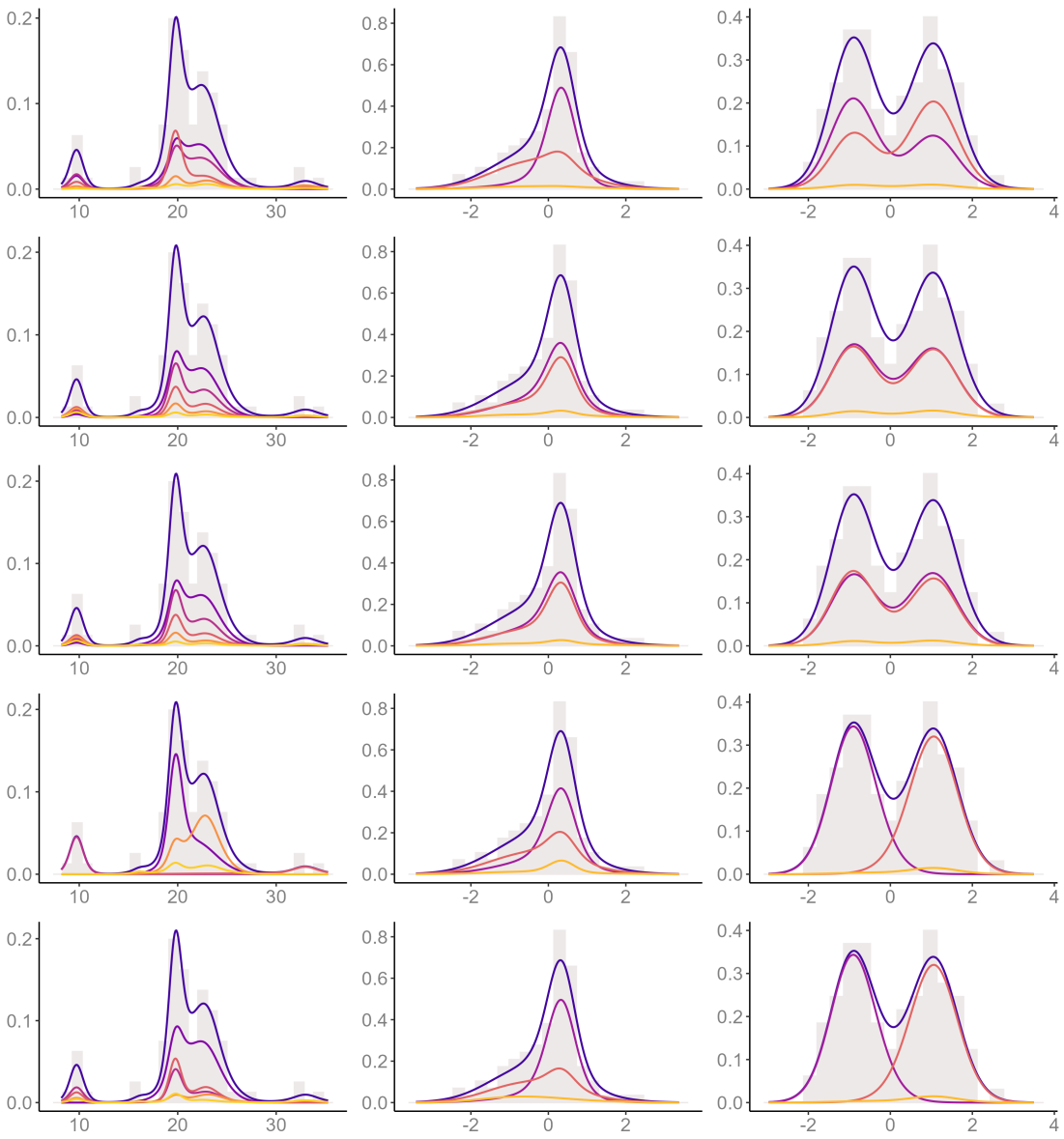

Figure E1: Estimated density and weighted densities by component (colored lines) for the galaxy (left column), leptokurtic (middle column) and bimodal (right column) data sets, assuming a Dirichlet process mixing distribution. Implementation was made using a Marginal sampler (1st row), the ordered allocation samplers in Sections 3.1 and 3.2 (2nd and 3rd row, respectively) including data permutations, the ordered allocation sampler in Section 3.1 without data permutations (4th row) and the slice-efficient sampler (5th row).

IAT of the deviance (D_v) and number occupied components (k_n) for these 5 different samplers. In Figure E1 we show the estimated weighted densities, $\widehat{Q}_j$, of each component $j \leq K$, as well as the estimated density, $\widehat{Q} = \sum_{j=1}^{K} \widehat{Q}_j$, where K is the largest index j, for which $\widehat{Q}_j$ is not identical to zero. We have computed

$$\widehat{Q}_j = \frac{1}{T} \sum_{t=1}^{T} \frac{n_j^{(t)}}{n} g(\cdot \mid x_j^{(t)}),$$

for a window of $T = 10^4$ iterations after the burn-in period has elapsed. Here $x_1^{(t)}, x_2^{(t)}, \ldots$ are the sampled component parameters as labelled (or indexed) at iteration t, and $n_j^{(t)}$ is the number of data points assigned to the jth component at iteration t. In particular, for the ordered allocation samplers, components are labelled in the order in which they were discovered by the (possibly permuted) dataset at each iteration. For the marginal sampler, components are labelled this way at the first iteration, and at subsequent iterations relabelling only occurred to delete gaps, i.e. so that the first k_n indexes, j, refer to the observed components. As for the slice sampler, components were never relabelled.

We will first focus on exploring the effects of the acceleration step in Section 3.3. As can be observed in Figure E1 the ordered allocation sampler without data permutations (OAS1*, 4th row) is more prone to label components consistently throughout iterations, this is reflected through the propensity of $\widehat{Q}_j$ to be unimodal. In contrast, for the ordered allocation sampler that includes the acceleration step of Section 3.3 (OAS1, 2nd row) $\widehat{Q}_j$ has the shape of the estimated density $\widehat{Q}$. Thus $\widehat{Q}_j$ is multimodal when $\widehat{Q}$ is (as is the case of the leptokurtic dataset). This indicates that more label-switches occur in the OAS1, which is an expected consequence of the fact that by permuting data points, components are discovered in a distinct order. Comparing against the graphs of the marginal sampler (1st row), we see that by including the acceleration step, label-switches occur in a more similar way as they naturally do in marginal samplers, which is the only type of sampler where labels are completely irrelevant. In terms of algorithmic performance, in Table E1 we see that the inclusion of data permutations at each iteration represents a significant improvement of the mixing properties, as the IAT corresponding to the OAS1 are much smaller than those of the OAS1*. The one exception is the IAT of D_v for the leptokurtic dataset, which is very similar for the OAS1 and the OAS1*. This is due to the fact that, although this dataset comes from more than one Gaussian component, it only has one mode. Thus, it is not clear from which component does each data point come from, this is turn leads to frequent label-switches even if one does not permute the dataset.

We now turn to explore the algorithmic performance of the OAS2 compared against that the OAS1, when the latter applies, i.e. the distribution of the size-biased permuted weights, $(\tilde{p}_j)$ is available. In Table E1 we see that for the galaxy dataset, the IAT values of the OAS2 compare very well with those of the OAS1. For the other two datasets instead there is a difference between the IAT values of the OAS1 and the OAS2. To explain why this happens, recall that the key distinction between the OAS1 and the OAS2 is that the first one updates $(\tilde{p}_j)$ directly, while the OAS2 relies on the the indexes (α_j). In particular for the DP model, the OAS2 ignores the fact that the distribution of $(\tilde{p}_j)$ is available. As mentioned in the main paper, to update (α_j) the OAS2 first updates $(\alpha_j)_{j \leq k_n}$ (i.e. those of weights of occupied components) and later $(\alpha_j)_{j > k_n}$, hence swaps between α_j and α_i, with $j \leq k_n < i$, mainly occur when occupied components are created or removed as a consequence of an update of (d_i). Now, the leptokurtic and bimodal datasets were both simulated from a mixture of two (more or less) balanced Gaussian

components. Since we have implemented a Gaussian mixture, at most iterations the sampler will effectively recognize that there are only two large occupied components (see the 2nd and 3rd columns of Figure E1 for an illustration). Thus the mixing of (α_j) will be affected because components are rarely created of removed. Furthermore, as typically there will be only two large "occupied" weights and the rest of them will be very small, it is extremely unlikely that their indexes get swapped. Instead, the galaxy dataset was not generated from a Gaussian mixture, which forces the model to rely on different Gaussian components of varying sizes to estimate the density (cf. 1st column of Figure E1). This means that the number of occupied components will change frequently and some of the "occupied" weights will be small at many iterations thus facilitating the mixing of weights' indexes.

		Marginal	OAS1	OAS2	OAS1*	Conditional
galaxy	D_v	12.30(0.23)	23.76(0.57)	26.76(0.84)	106.9(7.98)	119.2(9.95)
	k_n	13.68(0.25)	32.49(0.81)	36.36(1.09)	115.2(7.18)	190.2(16.3)
leptokurtic	D_v	22.42(0.54)	25.81(0.63)	35.98(1.00)	26.92(0.75)	120.5(10.3)
	k_n	9.26(0.13)	18.99(0.41)	27.97(0.88)	44.48(0.60)	100.8(7.18)
bimodal	D_v	7.84(0.16)	13.87(0.35)	20.82(0.57)	50.47(3.20)	35.61(1.68)
	k_n	6.30(0.07)	13.38(0.22)	25.43(0.87)	39.28(1.33)	52.00(2.18)

Table E1: IAT of D_v and k_n (standard errors are shown in parenthesis) for Algorithm 8 in Neal (2000) (Marginal), the ordered allocation samplers in Sections 3.1 and 3.2 with data permutation (OAS1 and OAS2, respectively), the ordered allocation sampler in Section 3.1 without data permutations (OAS1*) and the dependent slice-efficient sampler (Conditional), obtained by fitting a DP model to the galaxy, leptokurtic and bimodal datasets.