On Monte Carlo methods for the Dirichlet process mixture model, and the selection of its precision parameter prior

CARLO VICENTINI

On Monte Carlo methods for the Dirichlet process mixture model, and the selection of its precision parameter prior

Carlo Vicentini

A Thesis presented for the degree of
Doctor of Philosophy

Department of Mathematical Sciences
Durham University
United Kingdom

September 2022

On Monte Carlo methods for the Dirichlet process mixture model, and the selection of its precision parameter prior

Carlo Vicentini

Submitted for the degree of Doctor of Philosophy

September 2022

Abstract:

Two issues commonly faced by users of Dirichlet process mixture models are: 1) how to appropriately select a hyperprior for its precision parameter α, and 2) the typically slow mixing of the MCMC chain produced by conditional Gibbs samplers based on its stick-breaking representation, as opposed to marginal collapsed Gibbs samplers based on the Polya urn, which have smaller integrated autocorrelation times.

In this thesis, we analyse the most common approaches to hyperprior selection for α, we identify their limitations, and we propose a new methodology to overcome them.

To address slow mixing, we revisit three label-switching Metropolis moves from the literature (Hastie et al., 2015; Papaspiliopoulos & Roberts, 2008), improve them, and introduce a fourth move. Secondly, we revisit two i.i.d. sequential importance samplers which operate in the collapsed space (Liu, 1996; S. N. MacEachern et al., 1999), and we develop a new sequential importance sampler for the stick-breaking parameters of Dirichlet process mixtures, which operates in the stick-breaking space and which has minimal integrated autocorrelation time. Thirdly, we introduce

the i.i.d. *transcoding algorithm* which, conditional to a partition of the data, can infer back which specific stick in the stick-breaking construction each observation originated from. We use it as a building block to develop the *transcoding sampler*, which removes the need for label-switching Metropolis moves in the conditional stick-breaking sampler, as it uses the better performing marginal sampler (or any other sampler) to drive the MCMC chain, and augments its exchangeable partition posterior with conditional i.i.d. stick-breaking parameter inferences after the fact, thereby inheriting its shorter autocorrelation times.

Declaration

The work in this thesis is based on research carried out in the Department of Mathematical Sciences at Durham University. No part of this thesis has been submitted elsewhere for any degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

I would like to thank Durham University for the stimulating academic environment, and I am very grateful to my supervisor for our meetings and discussions.

Dedicated to

my parents and my wife

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Consider a Dirichlet process mixture model with random precision parameter α, inducing K_n clusters over n observations through its latent random partition. Such a model requires some distributional assumption for $p(\alpha \mid \boldsymbol{\eta})$ (the prior distribution of α), and it also requires the assignment of some fixed value to the parameter vector $\boldsymbol{\eta}$. Current common approaches either involve improper or quasi-degenerate priors, such as $\alpha \sim \text{Gamma}(a, b)$ with a and b close to zero, or methods which rely on the assessment of the distributional shape that $p(\alpha \mid \boldsymbol{\eta})$ would induce on the prior of K_n (Dorazio, 2009; Murugiah & Sweeting, 2012). Both methods have limitations, especially as n grows, which justify further study and a new proposal, which we articulate under the stick-breaking representation of the Dirichlet process.

Separately, inference of the Dirichlet process mixture model parameters under the stick-breaking representation is known to suffer from slowdowns caused by difficulties in the Gibbs sampler moving between local modes (Hastie et al., 2015; Papaspiliopoulos & Roberts, 2008). This class of samplers is known as *conditional* samplers, or *stick-breaking* samplers. Conversely, *marginal* Gibbs samplers (also called *collapsed* samplers) based on the Polya urn representation of the Dirichlet process operate in a smaller parameter space, as they marginalise the Dirichlet process out; they are

known to be less affected by the issue, and to have smaller integrated autocorrelation times. According to Papaspiliopoulos and Roberts, 2008, stick-breaking conditional samplers still have their advantages though, because they allow inference on the stick-breaking parameters, which the marginal samplers do not allow for, and because, under certain conditions, the probability distribution of their sticks can be modified to generalise them to a wider set of models than the Dirichlet process, such as for example the Pitman-Yor process, and obtain inferences accordingly. Therefore, various Metropolis jumps have been proposed to induce the stick-breaking sampler to move more frequently between local modes, thereby attaining faster mixing; however, none of these methodologies has so far reached the same level of performance as that of the marginal samplers, hence there is merit in attempting the design of more effective jumps.

Furthermore, given that the issue at hand is one pertaining to the autocorrelation of the posterior sample induced by the use of Gibbs, we turn our attention to sequential importance sampling, where a couple of promising studies was published in the late nineties (Liu, 1996; S. N. MacEachern et al., 1999), which had limited follow-up, while we see potential due to the fact that sequential importance samplers produce i.i.d. observations, hence they do not suffer from autocorrelation issues. These two samplers were devised before the stick-breaking representation of the Dirichlet process was widely introduced, hence their articulation only covers the Polya urn representation of the same, and we extend it to stick-breaking.

Finally, we observe a gap in literature pertaining to the study of the relationship between the collapsed parameter space of the Polya urn, and the wider parameter space of the stick-breaking representation. The former leads to cluster labels which are numbered in order of appearance, and which carry no specific meaning as they are exchangeable; the latter instead carry very specific meaning, as they point to which stick in the stick-breaking construction each observation originates from. We discuss the relationship between the two encodings, and we devise a way to infer one from the other, in both directions, to ultimately design a new sampler which

can augment the partition posterior obtained under any encoding, with inferences pertaining to all other stick-breaking parameters.

1.2 Contribution

Our main contributions to the topic of how to best assign a prior to α are as follows. First, we identify three main groupings to categorise common existing approaches into: sample-size-dependent, quasi-degenerate and improper priors. Second, we increase the class of sample-size-dependent priors by introducing Jeffreys' prior, and we derive a way to sample from it, which is non-trivial as its distribution has infinite mean. Its introduction is also useful to compare the prior that it implicitly induces on K_n with the assumptions that other sample-size-dependent priors make as to what a desirable prior distribution of K_n should be. Third, we introduce a new approach to the specification of $p(\alpha)$, which is independent of the sample size and which instead is based on the appraisal of the implied joint distribution of the stick-breaking weights (either in size-biased order, or ranked); we show an example of a modelling set-up where multiple DPMs stem from a common prior $p(\alpha; \boldsymbol{\eta})$, and where sample-size-dependent approaches are inapplicable while our sample-size-independent method is feasible.

To address the slow mixing issue of the stick-breaking sampler, we first revisit three Metropolis label-switching moves from literature: *move 1*, *move 2* and *move 3* (Hastie et al., 2015; Papaspiliopoulos & Roberts, 2008). We show how the published acceptance rate formula of move 3 was incomplete and would not lead to the correct posterior; we complete it and we also propose an improvement to both move 2 and move 3. We introduce move 4, which performs similarly to the improved version of move 3, and therefore constitutes a valid and, in our view, more intuitive alternative.

Secondly, to address the aforementioned slow mixing we develop a sequential importance sampler for the stick-breaking parameters of Dirichlet process mixtures, which produces i.i.d. samples and which therefore has minimal integrated autocorrelation

time. Although it suffers from sensitivity to the order whereby the observations appear in the data set (S. N. MacEachern et al., 1999), which needs to be monitored and which we know no optimality criterion for, we find its performance to be materially better than that of the slice sampler in all of the aspects we analyse.

Thirdly, we work out the relationship between the exchangeable cluster label encoding that arises from the Polya urn construction of the Dirichlet process, and the more informative cluster label encoding which arises from its stick-breaking construction. We develop an algorithm to infer the stick-breaking encoding from the Polya-urn exchangeable encoding. We ultimately develop the *transcoding sampler*, which removes the need for any label-switching Metropolis moves in the conditional stick-breaking sampler, as it uses the better performing marginal sampler (or any other sampler) to drive the MCMC chain, and augments its exchangeable partition posterior with i.i.d. stick-breaking parameter inferences after the fact, thereby inheriting its autocorrelation times.

1.3 Outline

Chapters 2 to 3 draw from the existing body of knowledge and set the basics and symbology for what follows.

Chapter 2 lays out the fundamentals of the Dirichlet process and of the Dirichlet process mixture, some of their many representations, and some of their properties, to support and justify subsequent results in later chapters. Chapter 3 summarises some key inferential algorithms for the Dirichlet process mixture model, and it covers three *collapsed*, or *marginal*, Polya-urn based samplers, and two stick-breaking, or *conditional*, samplers – namely the finite DP sampler and the slice sampler – which we use as building blocks and which we improve on in later chapters. Since one of our key contributions pertains to how to best decide the prior distribution $p(\alpha)$, we also cover some methodologies to sample from its posterior $p(\alpha \mid \boldsymbol{y})$.

Chapter 4 discusses methods to best decide the specification of $p(\alpha; \boldsymbol{\eta})$, the prior of the precision parameter of a Dirichlet process. Chapter 5 discusses label-switching moves to accelerate the stick-breaking slice sampler. Chapter 6 discusses two sequential importance samplers from the literature, and introduces and tests one new. Chapter 7 introduces the *transcoding sampler*. Conclusions and directions for future research are summarised in Chapter 8.

The test data set that we use throughout the thesis is the thumb tack data of Beckett and Diaconis, 1994; as the sequential importance sampler is sensitive to the order of the data points in the sample, it is worth mentioning that we use the data in the same order as it appears in Liu, 1996.

Chapter 2

Dirichlet process fundamentals

In this chapter we discuss the well-known Dirichlet process and Dirichlet process mixture model and some of their basic properties which are instrumental to later chapters.

2.1 The Dirichlet Process

Following the original definition by Ferguson, 1973, consider a space S and its σ-algebra $\mathcal{B}$, and define μ as a finite non-null measure on $(S, \mathcal{B})$. The stochastic process G, indexed by elements B of $\mathcal{B}$, is a Dirichlet Process with parameter μ if

$$G(B_1), \ldots, G(B_d) \sim \mathcal{D}(\mu(B_1), \ldots, \mu(B_d)),$$

for any measurable partition $(B_1, \ldots, B_d)$ of S, where $\mathcal{D}$ is the Dirichlet distribution.

G is an almost surely discrete random probability measure (Ferguson, 1973, page 218); its own probability distribution is typically indicated as $\pi(G)$. Often[1], μ is expressed as $\alpha G_0 \equiv \mu$, where $\alpha > 0$ and G_0 is a probability measure, and the Dirichlet Process is denoted either $\mathrm{DP}(\mu)$ or $\mathrm{DP}(\alpha, G_0)$. The latter notation, which we adopt, more easily allows to distinguish between G_0 (the expected value of G)

[1] See for example Hjort et al., 2010.

and its precision parameter α, which is so called because of its inverse relationship with the variance exhibited by G around its expected value G_0.

A model of n observations from a realisation of $G \sim \mathrm{DP}(\alpha, G_0)$ is usually written as follows, in hierarchical form:

$$\begin{aligned} \theta_i \mid G &\sim G, \quad i = 1, \ldots, n \\ G &\sim \mathrm{DP}(\alpha, G_0), \end{aligned} \tag{2.1.1}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ represents the observations, while G indicates both the Dirichlet Process and its realisation. The parameter α may also be assumed to be itself random, with prior distribution $p(\alpha \mid \boldsymbol{\eta})$; when so, the equations above become:

$$\begin{aligned} \theta_i \mid G, \alpha &\sim G \mid \alpha, \quad i = 1, \ldots, n \\ G \mid \alpha &\sim \mathrm{DP}(\alpha, G_0), \\ \alpha &\sim p(\alpha \mid \boldsymbol{\eta}). \end{aligned}$$

In the remainder of this thesis, we do not always explicitly condition on α in our formulae, for ease of notation.

As G_0 is the expected value of G, we have that, unconditional to G,

$$p(\theta_i) = \int G(\theta_i) \, \mathrm{d}\pi(G) = G_0(\theta_i), \quad i = 1, \ldots, n, \tag{2.1.2}$$

where π indicates the Dirichlet process probability measure of G.

Ferguson, 1973 also proved that the posterior distribution of a $\mathrm{DP}(\alpha, G_0)$ given n observations $(\theta_1, \ldots, \theta_n)$ is a $\mathrm{DP}\left(\alpha + n, \frac{\alpha G_0 + \sum_{i=1}^{n} \delta_{\theta_i}}{\alpha + n}\right)$; we observe that its expected value is the mixture distribution

$$\frac{\alpha}{\alpha + n} G_0(\theta_{n+1}) + \sum_{i=1}^{n} \frac{1}{\alpha + n} \delta_{\theta_i}(\theta_{n+1}),$$

meaning that, unconditional to G,

$$p(\theta_{n+1} \mid \theta_1, \ldots, \theta_n) = \frac{\alpha}{\alpha + n} G_0(\theta_{n+1}) + \sum_{i=1}^{n} \frac{1}{\alpha + n} \delta_{\theta_i}(\theta_{n+1}), \tag{2.1.3}$$

which echoes one of the main results from Blackwell and MacQueen, 1973.

Equations 2.1.2 and 2.1.3 marginalise G out and therefore they offer a way to draw a finite sample $\theta_1, \ldots, \theta_n$ from the hierarchical model of 2.1.1 without the need to sample the infinite-dimensional random probability measure G from $\mathrm{DP}(\alpha, G_0)$ first.

While the above is the most widely used and historically relevant definition of the Dirichlet Process, it is by no means the only one and in fact the Dirichlet Process can be defined or obtained in several other ways. In the following sections we only outline those which are instrumental to our results.

2.1.1 The Polya urn construction

Blackwell and MacQueen, 1973 considered a sequence $\{X_n, n \geq 1\}$ of random variables with values in S, where for every $B \subset S$:

$$p(X_1 \in B) = \mu(B)/\mu(S),$$
$$p(X_{n+1} \in B \mid X_1, \ldots, X_n) = \mu_n(B)/\mu_n(S),$$

and where $\mu_n := \mu + \sum_1^n \delta_{X_i}$ (see the parallel with equation 2.1.2 and 2.1.3 respectively).

They named it the *Polya sequence*, and they pointed out that, for finite S, it represents the results of successive draws from an urn which initially includes $\mu(x)$ balls of each colour $x \in S$ and, after each draw, the ball drawn is replaced and another ball of its same colour is added to the urn. In the infinite case (see Hoppe, 1984), the urn initially includes $\mu(S)$ black balls; in subsequent draws, if a black ball is drawn, it is returned to the urn and one additional ball of a previously unobserved colour is added in, otherwise if the ball drawn is not black it is returned to the urn, together with an additional ball of the same colour. In this case, the Polya sequence represents the colours of the balls added to the urn at each iteration, where any new *colour* corresponds in turn to a draw from S according to $\mu(B)/\mu(S)$.

They proved that:

1. $\mu_n/\mu_n(S)$ converges almost surely to a limiting discrete measure G with distribution $\mathrm{DP}(\mu)$, as $n \to \infty$;

2. conditional to G, the variables $X_1, X_2, \ldots$ are independent with distribution G.

The Polya Urn construction, which is fundamentally equivalent to equations 2.1.2 and 2.1.3 from Ferguson, 1973, therefore establishes a way to approximately sample the random probability measure G from $\mathrm{DP}(\alpha, G_0)$, and which consists in repeatedly sampling from 2.1.2 and 2.1.3 for n infinitely large, and then calculating the frequencies of the repeated values.

2.1.2 The stick-breaking construction

Sethuraman, 1994 considered

$$\begin{aligned} w_1 &:= v_1, \\ w_h &:= v_h \prod_{l<h} (1 - v_l), \quad h = 2, 3, \ldots \\ v_h &\sim \mathrm{Beta}(1, \alpha), \quad h = 1, 2, \ldots, \\ m_h &\sim G_0, \end{aligned} \tag{2.1.4}$$

where G_0 is a probability measure, and proved that the distribution of the random measure

$$G := \sum_{h=1}^{\infty} w_h \delta_{m_h} \tag{2.1.5}$$

is $DP(\alpha, G_0)$. The distribution of $\boldsymbol{w} := (w_1, w_2, \ldots)$ alone is known as the Griffiths–Engen–McCloskey distribution, and it is denoted by $\mathrm{GEM}(\alpha)$ (Arratia et al., 2003, section 4.8).

The equations above allow to directly sample a finite number of atoms of $G \sim DP(\alpha, G_0)$, in a way that is exact, unlike the Polya urn construction which instead relies on $n \to \infty$ and on the asymptotic build-up of the discrete probability masses due to observed ties in the data.

The probability distribution of the first $H-1$ elements of $\boldsymbol{w}$ can be derived from the H-dimensional generalised Dirichlet distribution (Connor & Mosimann, 1969) as follows. In the generalised Dirichlet distribution, the last element of vector $\boldsymbol{w}$ is defined as

$$w_H = 1 - \sum_{h=1}^{H-1} w_h,$$

and

$$v_h \sim \text{Beta}\left(a_h, b_h\right), \quad h = 1, \ldots, H-1.$$

Therefore, with the generalised Dirichlet distribution we have:

$$\begin{aligned} p\left(w_1, \ldots, w_{H-1}\right) = \left[\prod_{h=1}^{H-1} \frac{\Gamma\left(a_h + b_h\right)}{\Gamma\left(a_h\right)\Gamma\left(b_h\right)}\right] w_1^{a_1-1} \ldots w_{H-1}^{a_{H-1}-1} \left(1 - w_1 - \ldots - w_{H-1}\right)^{b_{H-1}-1} \\ \times \left(1 - w_1\right)^{b_1-(a_2+b_2)} \\ \times \left(1 - w_1 - w_2\right)^{b_2-(a_3+b_3)} \\ \times \ldots \\ \times \left(1 - w_1 - \ldots - w_{H-2}\right)^{b_{H-2}-(a_{H-1}+b_{H-1})}. \end{aligned} \tag{2.1.6}$$

Since in the Dirichlet process $v_h \sim \text{Beta}\left(1, \alpha\right)$, we have that $a_h = 1$ and $b_h = \alpha$, for $h = 1, 2, \ldots$, and equation 2.1.6 simplifies to the following:

$$p\left(w_1, \ldots, w_{H-1}\right) = \alpha^{H-1} \frac{\left(1 - w_1 - \ldots - w_{H-1}\right)^{\alpha-1}}{\left(1 - w_1\right)\left(1 - w_1 - w_2\right)\ldots\left(1 - w_1 - \ldots - w_{H-2}\right)}. \tag{2.1.7}$$

This result is due to the fact that, when $a_h = 1$ and $b_h = \alpha$, for $h = 1, 2, \ldots$, we have that

$$\prod_{h=1}^{H-1} \frac{\Gamma\left(a_h + b_h\right)}{\Gamma\left(a_h\right)\Gamma\left(b_h\right)} = \left(\frac{(1+\alpha-1)!}{(1-1)!\,(\alpha-1)!}\right)^{H-1} = \left(\frac{\alpha!}{(\alpha-1)!}\right)^{H-1} = \alpha^{H-1},$$

and all terms $w_1^{a_1-1} \ldots w_{H-1}^{a_{H-1}-1}$ in equation 2.1.6 are equal to 1 because all of their exponents are equal to 0.

Equation 2.1.7 applies to the Dirichlet process too, as well as to the generalised Dirichlet distribution, for any arbitrary $H-1$. By shifting the indices, equation 2.1.7 can be re-written to express the density of the first H weights of the Dirichlet

process as:

$$p\left(w_1, \ldots, w_H\right) = \alpha^H \frac{\left(1 - w_1 - \ldots - w_H\right)^{\alpha - 1}}{\left(1 - w_1\right) \ldots \left(1 - w_1 - \ldots - w_{H-1}\right)}.$$

Size-biased random permutations

A size-biased random permutation $\tilde{\boldsymbol{w}} = (\tilde{w}_1, \tilde{w}_2, \ldots)$ of $\boldsymbol{w} = (w_1, w_2, \ldots)$ is defined as follows (see for example Phadia, 2015). First sample $\tilde{w}_1$ from

$$p\left(\tilde{w}_1 = w_h \mid \boldsymbol{w}\right) = w_h, \quad h = 1, 2, \ldots,$$

and then sample $\tilde{w}_2, \tilde{w}_3, \ldots$ from

$$p\left(\tilde{w}_{k+1} = w_h \mid \tilde{w}_1, \ldots, \tilde{w}_k, \boldsymbol{w}\right) = \frac{w_h \, \mathbb{1}\left[w_h \neq \tilde{w}_i, \text{for } 1 \leq i \leq k\right]}{1 - \tilde{w}_1 - \tilde{w}_2 - \ldots - \tilde{w}_k}, \; h = 1, 2, \ldots \quad (2.1.8)$$

for $k \geq 1$.

An equivalent definition (Pitman, 1996a) involves sampling the indices $I_1, I_2, \ldots$ from the categorical distribution with parameter $\boldsymbol{w}$, and denoting the distinct values in $I_1, I_2, \ldots$, in order of appearance, as $\tilde{I}_1, \tilde{I}_2, \ldots$. Then $(\tilde{w}_1, \tilde{w}_2, \ldots) := \left(w_{\tilde{I}_1}, w_{\tilde{I}_2}, \ldots\right)$ is a size-biased permutation of $\boldsymbol{w}$.

An important property of $\boldsymbol{w}$ is that it is invariant to size-biased random permutations (Arratia et al., 2003, section 4.11; Pitman, 1996a, page 526; Engen, 1975; McCloskey, 1965), meaning that the random variable $\boldsymbol{w}$ and its size-biased random permutation $\tilde{\boldsymbol{w}}$ are equal in distribution:

$$\boldsymbol{w} \overset{d}{=} \tilde{\boldsymbol{w}}.$$

Because it has the same probability distribution as that of its random size-biased permutation, the sequence $w_1, w_2, \ldots$ is said to be *in size-biased order*. The random variable arising from any number of repeated size-biased random permutations of $\boldsymbol{w}$ still has the same probability distribution as $\boldsymbol{w}$.

2.1.3 The Poisson-Dirichlet Process representation

Kingman, 1975 introduced the Poisson-Dirichlet distribution (PDD), which he constructed through the gamma process $\xi(t)$, a stochastic process with $\xi(0) = 0$ and with increments which are independent on disjoint intervals, and gamma distributed. The PDD with parameter α is known to be equivalent to the distribution of the decreasing order statistics of $\boldsymbol{w}$ (Arratia et al., 2003, section 4.11)(Pitman, 1996a); we denote them by $\left(w_1^{\downarrow}, w_2^{\downarrow}, \ldots\right)$. Similarly, the decreasing weight-ranked equivalent of equation 2.1.5 is known as the Poisson-Dirichlet Process (PDP).

As laid out in Watterson, 1976, the conditional joint probability distribution of the first r ranked weights is:

$$p\left(w_1^{\downarrow}, \ldots, w_r^{\downarrow}\right) = \alpha^r \Gamma(\alpha)\, e^{\gamma\alpha} \frac{w_r^{\downarrow\alpha-1}}{w_1^{\downarrow} \cdots w_r^{\downarrow}}\; g\left(\frac{1 - w_1^{\downarrow} - \ldots - w_r^{\downarrow}}{w_r^{\downarrow}}\right), \tag{2.1.9}$$

where g is a function which can be recursively written as:

$$g(z) = z^{\alpha-1}\left[g(n)\, n^{1-\alpha} - \alpha \int_n^z g(y-1)\, y^{-\alpha} dy\right], \quad n \leq z < n+1,\ n := \lfloor z \rfloor,$$

and which is known to be particularly difficult to compute.

2.1.4 Properties

Since G is countably infinite and almost surely discrete, we have that $\theta_i = \theta_j$ with probability 1, for some $i \neq j$. Repeated values in $\boldsymbol{\theta}$ induce implicit clustering on $\boldsymbol{\theta}$ and, in turn, a random partition model (see section 2.2.1).

This can be observed in equation 2.1.3, where if G_0 is continuous, we obtain

$$p\left(\theta_i \notin \{\theta_1, \ldots, \theta_{i-1}\} \mid \alpha\right) = \frac{\alpha}{\alpha + i - 1},\ i = 2, \ldots, n,$$

hence the random number of clusters $K_n \mid \alpha$ observed in a sample of n observations is distributed as the sum of n Bernoulli variables with parameters $p_i = \alpha/(\alpha + i - 1)$.

Further,

$$\mathbb{E}\left[K_n \mid \alpha\right] = \sum_{i=1}^{n} \frac{\alpha}{\alpha + i - 1} \sim \alpha \log \frac{n+\alpha}{\alpha} \sim \alpha \log n, \tag{2.1.10}$$

$$\mathbb{V}\mathrm{ar}\left[K_n \mid \alpha\right] = \sum_{i=1}^{n} \frac{\alpha\,(i-1)}{(\alpha + i - 1)^2} \sim \alpha \log n, \tag{2.1.11}$$

as $n \to \infty$ (Antoniak, 1974; Arratia et al., 2003). The probability distribution of $K_n \mid \alpha$ is:

$$p\left(K_n = k \mid \alpha\right) = s_{n,k}\ \alpha^k\ \frac{\Gamma\left(\alpha\right)}{\Gamma\left(\alpha + n\right)}, \tag{2.1.12}$$

where $s_{n,k}$ are unsigned Stirling numbers of the first kind. The parameter α therefore critically controls $p\left(K_n \mid \alpha\right)$: we show in sections 4.2.2 and 4.2.3 that, for $\alpha \to 0$, the random variable $K_n \mid \alpha$ converges to the Dirac measure δ_1, while for $\alpha \to \infty$, the same converges to δ_n; similarly, if α is random, $p\left(\alpha \mid \boldsymbol{\eta}\right)$ determines the prior $p\left(K_n\right)$ through mixing, as

$$p\left(K_n = k\right) = \int_0^\infty s_{n,k}\ \alpha^k\ \frac{\Gamma\left(\alpha\right)}{\Gamma\left(\alpha + n\right)} \mathrm{d}p\left(\alpha \mid \boldsymbol{\eta}\right). \tag{2.1.13}$$

The following also holds (Arratia et al., 2000; Watterson, 1974):

$$d_{TV}\left(p\left(K_n \mid \alpha\right), \mathrm{Po}\left(\mathbb{E}\left[K_n \mid \alpha\right]\right)\right) = O\left(\frac{1}{\log n}\right), \tag{2.1.14}$$

where d_{TV} indicates total variation distance and $\mathrm{Po}\left(\lambda\right)$ indicates the Poisson distribution with parameter λ, and which implies convergence in distribution since $1/\log n \to 0$ for $n \to \infty$. While the relationship above is unlikely to carry practical use in computations, as its rate of convergence is sublinear, it does provide conceptual insight into the limiting behaviour of $p\left(K_n \mid \alpha\right)$, as we show in section 4.2.1.

2.2 The Dirichlet Process Mixture

Mixing the almost surely discrete Dirichlet Process with a parametric likelihood $p\left(\cdot \mid \theta\right)$ leads to what is known as a Dirichlet Process Mixture (DPM) (Ferguson,

1983; Lo, 1984), which in mixture form can be written as

$$p(y_i \mid G) = \int p(y_i \mid \theta_i)\, dG(\theta_i), \tag{2.2.1}$$
$$G \sim \mathrm{DP}(\alpha, G_0),$$

while in hierarchical form it can be written as

$$y_i \mid \theta_i \sim p(y_i \mid \theta_i), \tag{2.2.2}$$
$$\theta_i \mid G \sim G,$$
$$G \sim \mathrm{DP}(\alpha, G_0),$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)$ is a vector of n observations while $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ is the vector of their latent parameters. Typical applications of the DPM are:

- flexible modelling of continuous distributions, where the almost surely discrete nature of the Dirichlet Process would be undesirable unless mixed with a parametric continuous distribution;
- latent cluster modelling, where the number of latent clusters associated with n observations is assumed to grow with n, potentially up to infinity.

More generally, DPMs arise whenever a Dirichlet Process is used as a prior in a hierarchical Bayesian setting, leading to semiparametric model specifications having a parametric likelihood and a nonparametric prior.

2.3 Random partitions and exchangeability

We complete this chapter with some definitions pertaining to partitions, for later use in this thesis. This section mainly draws its definitions from Pitman, 1995.

Define an *integer partition* of $n \in \mathbb{N}^+$ as an unordered set $\{n_1, \ldots, n_k\}$ of positive integers such that $\sum_{i=1}^{k} n_i = n$. Define a composition of n as an ordered partition $(n_1, \ldots, n_k)$ of n. Similarly, define a *set partition* $\{A_1, \ldots, A_k\}$ of $\mathbb{N}_n = \{1, \ldots, n\}$

as an unordered collection of disjoint non-empty subsets of $\mathbb{N}_n$, with $\cup_i A_i = \mathbb{N}_n$; observe that its cardinalities $n_i = |A_i|$ induce an integer partition of n as defined above.

Define a *random partition* of n or of $\mathbb{N}_n$ as a random variable with values in the set of all partitions of n or $\mathbb{N}_n$ respectively. Say that a random partition of $\mathbb{N}_n$ is *exchangeable* if its distribution is invariant to all permutations of $\mathbb{N}_n$; say it is *partially exchangeable* if, for every partition $\{A_1, \ldots, A_k\}$, where the $A_1, \ldots, A_k$ are in order of appearance, their probability distribution is given by

$$p\left(|A_1|, \ldots, |A_k|\right) = p\left(n_1, \ldots, n_k\right), \tag{2.3.1}$$

and where by *order of appearance* we mean that $1 \in A_1$, and for each $2 \leq i \leq k$, the first element of $\mathbb{N}_n \setminus (A_1 \cup \ldots \cup A_{i-1})$ belongs to A_i. It can be shown that a random partition is exchangeable if and only if its probability distribution from equation 2.3.1 is symmetric for every permutation σ of $\mathbb{N}_k$:

$$p\left(n_1, \ldots, n_k\right) = p\left(n_{\sigma_1}, \ldots, n_{\sigma_k}\right). \tag{2.3.2}$$

For completeness, we also mention a different notion of exchangeability, which pertains to random vectors rather than to random partitions (de Finetti, 2011). A random vector $(x_1, \ldots, x_n)$ is exchangeable if

$$p\left(x_1, \ldots, x_n\right) = p\left(x_{\sigma_1}, \ldots, x_{\sigma_n}\right),$$

for any permutation σ of $\mathbb{N}_n$; a random vector $(x_, \ldots, x_n, y_1, \ldots, y_m)$ is partially exchangeable if

$$p\left(x_1, \ldots, x_n, y_1, \ldots, y_m\right) = p\left(x_{\sigma_1}, \ldots, x_{\sigma_n}, y_{\tau_1}, \ldots, y_{\tau_m}\right),$$

for any permutation σ of $\mathbb{N}_n$ and τ of $\mathbb{N}_m$.

2.3.1 EPPF and Ewens' sampling formula

As per equation 2.3.2, an exchangeable random partition Π_n can be described by the following function, which is called *exchangeable partition probability function* (EPPF):

$$p(n_1, \ldots, n_k) = \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \prod_{j=1}^{k} \Gamma(n_j). \tag{2.3.3}$$

The EPPF is a symmetric function which returns the probability of one particular (unordered) set partition, and which only depends on the unordered block sizes; partitions with the same block sizes have the same probability.

Equation 2.3.3 is also related to the following, which is called the *Ewens sampling formula*. Define $M_j := \#\{i : n_i = j,\ i = 1, \ldots, k\},\ j = 1, \ldots, n$. Then

$$p(M_1, \ldots, M_n) = n! \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \prod_{i=1}^{n} \frac{\alpha^{M_i}}{i^{M_i} M_i!}, \tag{2.3.4}$$

as for every configuration in Equation 2.3.4 there are

$$\frac{n!}{\prod_{i=1}^{n} i!^{M_i} M_i!}$$

configurations in 2.3.3 (see Crane, 2016 for a comprehensive summary of the uses of the Ewens sampling formula).

If instead of partitions, one is interested in compositions of n, then the probability of the composition $(n_1, \ldots, n_k)$ in order of appearance is (Pitman, 2002, equation 2.6):

$$\begin{aligned} p_{\text{OOA}}(n_1, \ldots, n_k) &= \frac{n!}{n_k (n_k + n_{k-1}) \cdots (n_k + \cdots + n_1) \prod_{i=1}^{k} (n_i - 1)!}\, p(n_1, \ldots, n_k) \\ &= \frac{n!}{n_k (n_k + n_{k-1}) \cdots (n_k + \cdots + n_1)}\, \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)}. \end{aligned} \tag{2.3.5}$$

Chapter 3

Inference on Dirichlet Process Mixtures

The goal of performing inference on the DPM is typically to obtain information on the posterior distributions of the unobservable parameters such as $\boldsymbol{\theta}, \boldsymbol{w}, \ldots$, or on their posterior predictive distribution, or again on the posterior predictive distribution of y_{n+1}. In this chapter we summarise some inferential methods based on Gibbs sampling (Geman & Geman, 1984), for subsequent use in the following chapters. We focus on three collapsed Gibbs samplers (also called *marginal* samplers), based on the Polya urn representation of the DP, and on two conditional samplers based on its stick-breaking representation.

3.1 Gibbs sampling

Gibbs sampling (Gelfand & Smith, 1990; Geman & Geman, 1984) is attractive in Bayesian statistics because it allows to approximately obtain very complex joint posterior distributions by simulating from a more accessible sequence of simpler marginal posteriors. In the limit, their sampling distribution converges to that of the posterior of interest.

More broadly, posterior samples can be used for inference as follows. Define $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(B)}$ as B posterior samples drawn from target density $p(\boldsymbol{\theta} \mid \boldsymbol{y})$. Because

$$\boldsymbol{\theta}^{(b)} \xrightarrow{d} \boldsymbol{\theta} \mid \boldsymbol{y} \tag{3.1.1}$$

and

$$\lim_{B \to \infty} \frac{1}{B} \sum_{b=1}^{B} t\left(\boldsymbol{\theta}^{(b)}\right) = E\left[t\left(\boldsymbol{\theta} \mid \boldsymbol{y}\right)\right], \tag{3.1.2}$$

for any measurable function t of $\boldsymbol{\theta} \mid \boldsymbol{y}$, the posterior samples can be averaged, to approximate the expected value of $\boldsymbol{\theta} \mid \boldsymbol{y}$:

$$E\left[\boldsymbol{\theta} \mid \boldsymbol{y}\right] = \int \theta \; dp\left(\boldsymbol{\theta} \mid \boldsymbol{y}\right) \approx \frac{1}{B} \sum_{b=1}^{B} \boldsymbol{\theta}^{(b)}. \tag{3.1.3}$$

Since multivariate convergence in distribution also implies element-by-element convergence in distribution, we also have that:

$$E\left[\theta_i \mid \boldsymbol{y}\right] \approx \frac{1}{B} \sum_{b=1}^{B} \theta_i^{(b)}. \tag{3.1.4}$$

Other statistics of interest can be obtained from the empirical distribution of the Gibbs samples.

Many inferential algorithms have been developed over the years to sample from the DPM posterior since its introduction, even in recent times. Some examples are:

- *collapsed* Gibbs samplers use the Polya urn representation to integrate G out, thereby avoiding the problem of having to draw samples from the infinite-dimensional random probability measure G;
- the *no-gaps* algorithm (S. MacEachern & Müller, 1998), the *complete model* algorithm (S. MacEachern & Müller, 1998, 2012) and their variants (Neal, 2000) (which are still *collapsed* algorithms) improve on the above, and allow to deal with the case where G_0 and the likelihood $p(y_i \mid \theta_i)$ are non-conjugate, by augmenting $\boldsymbol{\theta}^{\star}$ with additional elements corresponding to empty clusters;
- other samplers rely on the stick breaking representation instead. Amongst them, *blocked* Gibbs samplers approximate G by truncating it to a finite number

of components, and by sampling blocks of variables from their multivariate distribution (Ishwaran & James, 2001; Ishwaran & Zarepour, 2000);

- *retrospective* (Papaspiliopoulos & Roberts, 2008) and *sliced* Gibbs samplers (Kalli et al., 2011; Walker, 2007) are also based on stick breaking, but entail no approximation error caused by truncation; this is accomplished by only drawing the atoms of G which are strictly necessary for inference, and by introducing auxiliary latent variables;

- many more samplers have been devised, not necessarily Gibbs, and amongst the many we mention *reversible jump* MCMC samplers and split-and-merge algorithms (Dahl & Newcomb, 2022; Jain & Neal, 2004), sequential imputation samplers (also called sequential importance sampling) (Liu, 1994, 1996; S. N. MacEachern et al., 1999), and variational Bayes samplers (Blei & Jordan, 2006).

In what follows, we outline some examples of collapsed Gibbs samplers, blocked Gibbs samplers, and slice samplers, which we improve on in subsequent chapters of this thesis.

3.2 Notation

We broadly follow the notation of Müller et al., 2015, section 2.3, which we extend to the stick breaking representation too. We introduce cluster membership indicators $\boldsymbol{s} := (s_1, s_2, \ldots)$ and $\boldsymbol{r} := (r_1, r_2, \ldots)$ to indicate which cluster of G each data point $(y_1, y_2, \ldots)$ originates from. We adopt two different encodings:

- in the Polya urn construction, we label clusters according to the sequence $1, 2, \ldots$ to reflect their order of appearance in the data, and we denote cluster membership by s_i, for $i = 1, 2, \ldots$. In this notation we always have $s_1 = 1$, and $s_i \leq i$, by definition. We also denote the distinct values of $\boldsymbol{\theta}$ by $\boldsymbol{\theta}^\star$, with $\theta_j^\star$ representing the value of the latent parameter in cluster j;

- conversely, in the stick breaking construction, we denote cluster membership by r_i to indicate which of the atoms $h = 1, 2, \ldots$ of G in equation 2.1.5 each data point y_i originates from. Contrary to s_1, in this setting r_1 is not restricted to any specific value of $\mathbb{N}^+$.

With reference to a finite sample of n observations, denote by k_i the number of unique elements in $(\theta_1, \ldots, \theta_i)$, for $i \leq n$. Let n_j indicate the number of observations belonging to cluster $j \in \{1, \ldots, k_n\}$, and $n_{i,j}$ the number of observations belonging to cluster $j \in \{1, \ldots, k_i\}$:

$$n_j = \sum_{l=1}^{n} \mathbb{1}_{\theta_l = \theta_j^\star},$$
$$n_{i,j} = \sum_{l=1}^{i} \mathbb{1}_{\theta_l = \theta_j^\star}.$$

Use the $^-$ superscript to indicate the same quantity measured on the data set after observation i is removed. For example:

$$n_j^- = \sum_{l \neq i} \mathbb{1}_{\theta_l = \theta_j^\star},$$

while k^- denotes the number of clusters in the reduced data set $\boldsymbol{y} \setminus \{y_i\}$. Use the subscript $_{-i}$ in relation to $\boldsymbol{\theta}$, $\boldsymbol{y}$ and $\boldsymbol{s}$ to indicate $\boldsymbol{\theta} \setminus \{\theta_i\}$, $\boldsymbol{y} \setminus \{y_i\}$ and $\boldsymbol{s} \setminus \{s_i\}$ respectively; $\boldsymbol{\theta}^{\star -}$ therefore represents the distinct values of $\boldsymbol{\theta}_{-i}$. Finally, denote by $\boldsymbol{y}_j^\star$ the set of observations $\{y_i : s_i = j\}$.

3.3 Collapsed Gibbs samplers

Collapsed Gibbs samplers revolve around the idea of marginalising G out, to avoid directly sampling from it, as G is infinite-dimensional[1]. In the following sections, we outline three variations.

Collapsed algorithm 1 first appeared in Escobar's PhD thesis (Escobar, 1988), and was later published as a journal article (Escobar, 1994). The algorithm sequentially

[1]For a thorough assessment of *collapsing* and *blocking* in the context of the Gibbs sampler, see van Dyk and Park, 2008 and Liu et al., 1994.

updates each $\theta_i \mid \ldots$ in isolation, one at a time. Cluster membership can be inferred by looking at repeated values in $\boldsymbol{\theta}$. Because each parameter is updated in isolation, and it depends on all the others, it may get stuck for a long time in a local high probability space. For faster mixing, Bush and MacEachern, 1996 proposed to optionally update $\boldsymbol{\theta}^\star$ at the end of each Gibbs cycle.

Collapsed algorithm 2 (S. N. MacEachern, 1994) re-parameterises $\boldsymbol{\theta}$ into two separate vectors $\boldsymbol{\theta}^\star$ and $\boldsymbol{s}$. In its first step, it marginalises $\boldsymbol{\theta}$ out, so that the cluster membership indicators $\boldsymbol{s}$ can be sampled directly, without necessarily dealing with the locations of the clusters $\boldsymbol{\theta}^\star$; this is aimed at reducing autocorrelation in the MCMC chain and to allow faster exploration of the parametric space. Then, in a second step it samples $\boldsymbol{\theta} \mid \boldsymbol{s}, \boldsymbol{y}$.

Collapsed algorithm 3 (Bush & MacEachern, 1996) is similar to algorithm 2, minus the extra marginalisation of $\boldsymbol{\theta}$ in the first step.

Algorithm 1, 2 and 3 above are also summarised in Neal, 2000, where they are referred to as algorithm 1, algorithm 3 and algorithm 2 respectively.

3.3.1 Collapsing the parameter space

All three algorithms above are *collapsed*, in the sense that they collapse the parametric space by integrating G out. First, notice that the specification of the model in equation 2.1.1 implies that the conditional joint probability of $\boldsymbol{\theta} \mid G$ is

$$p(\theta_1, \ldots, \theta_n \mid G) = \prod_{i=1}^{n} G(\theta_i), \tag{3.3.1}$$

and that in turn its unconditional joint probability is

$$p(\theta_1, \ldots, \theta_n) = \int \prod_{i=1}^{n} G(\theta_i) \, \mathrm{d}\pi(G). \tag{3.3.2}$$

Collapsed Gibbs samplers move from equation 3.3.2, rather than from 3.3.1. By the chain rule of probability, equation 3.3.2 can be written as:

$$p(\theta_1, \ldots, \theta_n) = p(\theta_n \mid \theta_{n-1}, \ldots, \theta_1) \, p(\theta_{n-1}, \ldots, \theta_1)$$

$$= p(\theta_n \mid \theta_{n-1}, \ldots, \theta_1)\, p(\theta_{n-1} \mid \theta_{n-2}, \ldots, \theta_1)\, p(\theta_{n-2}, \ldots, \theta_1)$$
$$= p(\theta_1) \prod_{i=2}^{n} p(\theta_i \mid \theta_{i-1}, \ldots, \theta_1),$$

and, because of equation 2.1.3, we obtain (Blackwell & MacQueen, 1973):

$$p(\theta_1, \ldots, \theta_n) = \int \prod_{i=1}^{n} G(\theta_i)\, d\pi(G)$$
$$= G_0(\theta_1) \prod_{i=2}^{n} \left(\frac{1}{i-1+\alpha} \sum_{j=1}^{i-1} \delta_{\theta_j}(\theta_i) + \frac{\alpha}{i-1+\alpha} G_0(\theta_i) \right). \quad (3.3.3)$$

3.3.2 Collapsing further

The vector $\boldsymbol{\theta}$ can be re-encoded into $(\boldsymbol{s}, \boldsymbol{\theta}^\star)$, to express the same information as $\boldsymbol{\theta}$, leading to the factorisation

$$p(\boldsymbol{\theta}) = p(\boldsymbol{s}, \boldsymbol{\theta}^\star) = p(\boldsymbol{s}) \cdot p(\boldsymbol{\theta}^\star \mid \boldsymbol{s}).$$

Because the parameter space of the posterior of $\boldsymbol{s}$ is smaller than that of $\boldsymbol{\theta}$, some collapsed samplers, such as algorithm 2, adopt this blocking scheme on the assumption that it is more efficient than sampling from the posterior of $\boldsymbol{\theta}$ (also note that sampling from the posterior of $\boldsymbol{\theta}^\star \mid \boldsymbol{s}$ is optional, if one is not interested). Therefore, this class of samplers collapses the parameter space even further than those outlined in section 3.3.1, as it focuses on the posterior of $\boldsymbol{s}$ rather than from that of $\boldsymbol{\theta}$. A priori, $p(\boldsymbol{s})$ is given by equation 2.3.3.

Collapsing and conjugacy

Collapsed algorithms can be further subdivided into those which are best suited for cases where G_0 and $p(y_i \mid \theta_i)$ are conjugate, and those which instead are best suited for cases where they are not.

Collapsed algorithms 1, 2 and 3 all use the following expression:

$$\int p(y_i \mid \theta)\, \mathrm{d}G_0(\theta), \quad (3.3.4)$$

and additionally algorithms 2 and 3 also require sampling from equation 3.3.8; both expressions have closed form when G_0 and $p(y_i \mid \theta_i)$ are conjugate, and when so, they are easy to implement, and efficient. On the other hand, these algorithms become considerably less attractive in the non-conjugate case; while lack of conjugacy does not invalidate them, and expressions 3.3.4 and 3.3.8 can still be evaluated with numerical integration, doing so carries high computational cost due to the overhead that the numerical integration requires at each Gibbs iteration.

The non-conjugate case is better handled with alternative collapsed algorithms, such as the *no-gaps* algorithm (S. MacEachern & Müller, 1998), or algorithm 8 from Neal, 2000. In particular, algorithm 8 is similar to collapsed algorithm 3, except that it uses an augmentation method which involves sampling m extra parameters at each iteration, and which avoids calculating expression 3.3.4 or sampling from equation 3.3.8, therefore not incurring into the aforementioned issues. As $m \to \infty$, the behaviour of algorithm 8 approaches that of collapsed algorithm 3, although at extra computational cost. More generally, non-conjugate algorithms are less efficient than their conjugate counterparts, except in the presence of a non-conjugate pair, when they are superior.

Because in this thesis our test data set (Beckett & Diaconis, 1994) does naturally lead to the use of a conjugate pair, we focus on collapsed algorithms 1, 2 and 3, and there is no need for the *no-gaps* algorithm or for Neal's algorithm 8.

3.3.3 Algorithm 1

The distribution $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ is obtained with the Gibbs sampler by repeatedly sampling from:

$$\begin{aligned} p(\theta_i \mid \boldsymbol{\theta}_{-i}, \boldsymbol{y}) &\propto p(\boldsymbol{y} \mid \boldsymbol{\theta})\, p(\theta_i \mid \boldsymbol{\theta}_{-i}) \\ &\propto p(y_i \mid \theta_i)\, p(\theta_i \mid \boldsymbol{\theta}_{-i}) \\ &\propto \alpha p(y_i \mid \theta_i)\, G_0(\theta_i) + \sum_{j \neq i} p(y_i \mid \theta_i)\, \delta_{\theta_j}(\theta_i), \end{aligned} \tag{3.3.5}$$

where the last passage is justified by equation 2.1.3.

The equation above identifies a mixture of distributions, because:

- $p(y_i \mid \theta_i) G_0(\theta_i)$ is the unnormalised posterior of $\theta_i \mid y_i$, and α is constant;
- $\delta_{\theta_j}(\theta_i)$ are Dirac measures, and their weights $p(y_i \mid \theta_i)$ are constant where the Dirac measures have non-zero mass.

It is therefore sufficient to re-express equation 3.3.5 as a weighted sum of probability measures by normalising $p(y_i \mid \theta_i) G_0(\theta_i)$:

$$\begin{aligned} p(\theta_i \mid \boldsymbol{\theta}_{-i}, y_i) &\propto \alpha p(y_i \mid \theta_i) G_0(\theta_i) + \sum_{j \neq i} p(y_i \mid \theta_i) \delta_{\theta_j}(\theta_i) \\ &\propto \alpha \int p(y_i \mid \theta) \, \mathrm{d}G_0(\theta) \frac{p(y_i \mid \theta_i) G_0(\theta_i)}{\int p(y_i \mid \theta) \, \mathrm{d}G_0(\theta)} + \sum_{j \neq i} p(y_i \mid \theta_i) \delta_{\theta_j}(\theta_i) \\ &\propto \alpha q_0 H(\theta_i \mid y_i) + \sum_{j \neq i} p(y_i \mid \theta_i) \delta_{\theta_j}(\theta_i), \end{aligned} \tag{3.3.6}$$

with

$$\begin{aligned} q_0 &= \int p(y_i \mid \theta) \, \mathrm{d}G_0(\theta), \\ H(\theta_i \mid y_i) &= \frac{p(y_i \mid \theta_i) G_0(\theta_i)}{\int p(y_i \mid \theta) \, \mathrm{d}G_0(\theta)}. \end{aligned}$$

Finally, equation 3.3.6 can be reassembled as:

$$\theta_i \mid \boldsymbol{\theta}_{-i}, y_i \sim \begin{cases} H(\theta_i \mid y_i), & \text{with unnormalised prob. } \alpha q_0, \\ \theta_j, \ j \neq i, & \text{with unnormalised prob. } p(y_i \mid \theta_j). \end{cases} \tag{3.3.7}$$

The normalising constant can then be easily calculated, if needed, as

$$c = \frac{1}{\alpha q_0 + \sum_{j \neq i} p(y_i \mid \theta_j)}.$$

For faster mixing, Bush and MacEachern, 1996 proposed to optionally update $\boldsymbol{\theta}^\star \mid \ldots$ at the end of each Gibbs cycle. At the end of each Gibbs iteration, a cluster structure $\boldsymbol{s} \mid \ldots$ is available; all that is needed is to sample from the following, for each

$j = 1, \ldots, k$:

$$p\left(\theta_j^\star \mid \boldsymbol{s}, \boldsymbol{y}\right) = p\left(\theta_j^\star \mid \boldsymbol{y}_j^\star\right) \propto G_0\left(\theta_j^\star\right) \prod_{\{i:s_i=j\}} p\left(y_i \mid \theta_j^\star\right), \tag{3.3.8}$$

which is the posterior of G_0, based on all the observations in $\boldsymbol{y}$ which belong to cluster j. In the conjugate case, this is an easy distribution to sample from.

3.3.4 Algorithm 2

Algorithm 2 replaces $\boldsymbol{\theta}$ with $(\boldsymbol{\theta}^\star, \boldsymbol{s})$, which provides the same information as $\boldsymbol{\theta}$, although with a different coding. This sampler proceeds in two steps, by drawing from $p(\boldsymbol{s} \mid \boldsymbol{y})$ and then drawing from $p(\boldsymbol{\theta}^\star \mid \boldsymbol{s}, \boldsymbol{y})$; both steps involve Gibbs sampling.

In the first step we sample from $p(s_i \mid \boldsymbol{s}_{-i}, \boldsymbol{y})$, for $i = 1, \ldots, n$. See that equation 3.3.7 implies:

$$p\left(s_i = j \mid \boldsymbol{s}_{-i}, \boldsymbol{y}, \boldsymbol{\theta}^{\star-}\right) \propto \begin{cases} \alpha q_0, & j = k^- + 1, \\ n_j^- p\left(y_i \mid \theta_i = \theta_j^{\star-}\right), & j = 1, \ldots, k^-. \end{cases} \tag{3.3.9}$$

Then, integrate $\boldsymbol{\theta}^{\star-}$ out:

$$p(s_i = j \mid \boldsymbol{s}_{-i}, \boldsymbol{y}) \propto \begin{cases} \alpha q_0, & j = k^- + 1, \\ n_j^- \int p(y_i \mid \theta) \, \mathrm{d}p\left(\theta \mid \boldsymbol{y}_j^{\star-}\right), & j = 1, \ldots, k^-, \end{cases} \tag{3.3.10}$$

which we can sample from, as it can be enumerated (as the number of possible values that s_i may assume is finite), and it can also be easily computed if $G_0(\theta)$ and $p(y \mid \theta)$ are conjugate.

In the second step, for each $j = 1, \ldots, k$, we sample $\theta_j^\star \mid \boldsymbol{s}, \boldsymbol{y}$ from 3.3.8.

3.3.5 Algorithm 3

Algorithm 3 draws $s_i \mid \boldsymbol{s}_{-i}, \boldsymbol{y}, \boldsymbol{\theta}^{\star-}$ for $i = 1, \ldots, n$ from equation 3.3.9, and $\theta_j^\star \mid \boldsymbol{s}, \boldsymbol{y}$ for $j = 1, \ldots, k$ from equation 3.3.8. In other words, algorithm 3 skips one of the

marginalisation steps of algorithm 2.

3.4 Stick-breaking Gibbs samplers

Contrary to collapsed Gibbs samplers, stick-breaking or blocked Gibbs samplers explicitly model G and do not marginalise it out. In doing so, they resort to *blocking* techniques, where several conditional random variables are drawn concurrently at a time, in *blocks*, from their multivariate distribution[2].

Here we discuss two such samplers: the finite DP approximation sampler (Ishwaran & James, 2001), and the slice sampler (Kalli et al., 2011; Papaspiliopoulos & Roberts, 2008; Walker, 2007).

3.4.1 Finite DP sampler

As one of the main difficulties of sampling from the stick-breaking representation DP is its infinite dimensionality, one possible solution is to approximate it with a finite dimensional construct. An easy way of doing so is by truncating the infinite summation over $h = 1, 2, \ldots$ in equation 2.1.5 to the first H summands (Ishwaran & James, 2001):

$$G \approx \sum_{h=1}^{H} w_h \delta_{m_h}, \tag{3.4.1}$$

which leads to the following DPM model specification:

$$\begin{aligned}
&p\left(y_i \mid \boldsymbol{w}, \boldsymbol{m}\right) = \sum_{h=1}^{H} w_h p\left(y_i \mid m_h\right), \\
&w_1 := v_1 \\
&w_h := v_h \left(\prod_{l<h} \left(1 - v_l\right)\right), \quad h = 2, \ldots, H \\
&v_h \sim \text{Beta}\left(1, \alpha\right), \quad h = 1, \ldots, H-1,
\end{aligned}$$

[2]For a thorough assessment of *collapsing* and *blocking* in the context of the Gibbs sampler, see van Dyk and Park, 2008 and Liu et al., 1994.

$$v_H = 1,$$
$$m_h \sim G_0.$$

With reference to the latent indicator r_i introduced in section 3.2, the following holds:

$$p(y_i \mid r_i = h, \boldsymbol{m}) = p(y_i \mid m_h),$$
$$p(r_i = h \mid \boldsymbol{w}) = w_h.$$

The finite DP sampler arrives at the joint probability $p(\boldsymbol{r}, \boldsymbol{w}, \boldsymbol{m} \mid \boldsymbol{y})$ by using a Gibbs scheme, where in turn it draws from $p(\boldsymbol{r} \mid \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{y})$ and from the joint distribution $p(\boldsymbol{w}, \boldsymbol{m} \mid \boldsymbol{r}, \boldsymbol{y})$. Since $\boldsymbol{w}, \boldsymbol{m}$ are conditionally independent, drawing from their joint distribution $p(\boldsymbol{w}, \boldsymbol{m} \mid \boldsymbol{r}, \boldsymbol{y})$ can be accomplished by separately sampling from $p(\boldsymbol{w} \mid \boldsymbol{r}, \boldsymbol{y})$ and $p(\boldsymbol{m} \mid \boldsymbol{r}, \boldsymbol{y})$. To summarise[3]:

1. sample from $p(\boldsymbol{r} \mid \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{y})$;
2. sample from $p(\boldsymbol{w} \mid \boldsymbol{r}, \boldsymbol{y})$;
3. sample from $p(\boldsymbol{m} \mid \boldsymbol{r}, \boldsymbol{y})$.

As with collapsed Gibbs, if α is random then an additional step can be introduced, to sample from its full conditional too.

In step 1, the algorithm draws each element in turn, conditional to all others:

$$\begin{aligned} p(r_i = h \mid r_{-i}, \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{y}) &= p(r_i = h \mid \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{y}) \\ &\propto p(r_i = h \mid \boldsymbol{w}, \boldsymbol{m}, y_i) \\ &\propto p(y_i \mid r_i = h, \boldsymbol{w}, \boldsymbol{m})\, p(r_i = h \mid \boldsymbol{w}, \boldsymbol{m}) \\ &\propto p(y_i \mid m_h)\, w_h, \quad i = 1, \ldots, n. \end{aligned}$$

In step 2, $\boldsymbol{w} \mid \boldsymbol{r}, \boldsymbol{y}$ is sampled from the joint conditional distribution $p(\boldsymbol{v} \mid \boldsymbol{r}, \boldsymbol{y})$,

[3]The order of the steps is interchangeable.

where $\boldsymbol{v} = (v_1, \ldots, v_{H-1})$. This is possible because $w_h,\ h = 1, \ldots, H$, is a function of elements of $\boldsymbol{v}$ (Connor & Mosimann, 1969).

Due to the conjugacy of the generalised Dirichlet distribution, the conditional distribution of $\boldsymbol{v} \mid \boldsymbol{r}$ is the product of

$$v_h \mid \ldots \sim \text{Beta}\left(1 + n_h, \alpha + \sum_{l=h+1}^{H} n_l\right), \tag{3.4.2}$$

where n_h is the number of observations in cluster h.

In step 3, values are drawn from

$$\begin{aligned} p\left(m_h \mid \boldsymbol{r}, \boldsymbol{y}\right) &\propto p\left(\boldsymbol{y} \mid m_h, \boldsymbol{r}\right) p\left(m_h \mid \boldsymbol{r}\right) \\ &\propto G_0\left(m_h\right) \prod_{\{i:r_i=h\}} p\left(y_i \mid m_h\right), \end{aligned}$$

for each h. Unoccupied clusters are sampled from G_0. Calculations are easy in the conjugate case.

3.4.2 Slice sampler

The slice sampler (Kalli et al., 2011; Papaspiliopoulos, 2008; Papaspiliopoulos & Roberts, 2008; Walker, 2007) builds on the stick-breaking representation and introduces latent random variables which allow simulation to be carried out over a finite number of components. While it leverages many aspects of the finite DP sampler, its truncation point is random and devised in such a way that the approach is exact; what differentiates it is also its blocking scheme. The variation that we discuss is based on Papaspiliopoulos, 2008; we do not discuss a later variation which can provide better performance but requires tuning (Kalli et al., 2011).

Given an observation y_i, introduce latent random variable u_i such that the joint density of y_i and u_i conditional to $\boldsymbol{w}, \boldsymbol{m}$ is:

$$\begin{aligned} p\left(y_i, u_i \mid \boldsymbol{w}, \boldsymbol{m}\right) &= \sum_{h=1}^{\infty} I\left(u_i < w_h\right) p\left(y_i \mid m_h\right) \\ &= \sum_{h \in A(u_i)} p\left(y_i \mid m_h\right), \end{aligned} \tag{3.4.3}$$

where $A(u_i) := \{h : w_h > u_i\}$. The latter is a finite set, due to the stick-breaking construction of $\boldsymbol{w}$. The joint density from equation 3.4.3 exists, because integrating u_i out gives back the conditional marginal distribution of $y_i \mid \boldsymbol{w}, \boldsymbol{m}$:

$$\int \sum_{h=1}^{\infty} I(u_i < w_h)\, p(y_i \mid m_h)\, d\mathcal{L}(u) = \sum_{h=1}^{\infty} w_h p(y_i \mid m_h) = p(y_i \mid \boldsymbol{w}, \boldsymbol{m}),$$

where $\mathcal{L}(\cdot)$ is the Lebesgue measure.

Notice that equation 3.4.3 can be rewritten as

$$p(y_i, u_i \mid \boldsymbol{w}, \boldsymbol{m}) = \sum_{h=1}^{\infty} w_h U_{0,w_h}(u_i)\, p(y_i \mid m_h), \tag{3.4.4}$$

where $U_{0,w_h}(\cdot)$ is the density of the uniform distribution on the $[0, w_h]$ interval. Equation 3.4.4 is a mixture where, with probability w_h, y_i and u_i are independent conditional to $\boldsymbol{w}, \boldsymbol{m}$, hence

$$p(y_i, u_i \mid \boldsymbol{w}, \boldsymbol{m}, r_i) = U_{0,w_{r_i}}(u_i)\, p(y_i \mid m_{r_i}),$$

which in turn leads to

$$\begin{aligned} p(y_i, u_i, r_i \mid \boldsymbol{w}, \boldsymbol{m}) &= p(r_i \mid \boldsymbol{w}, \boldsymbol{m}) \cdot p(y_i, u_i \mid \boldsymbol{w}, \boldsymbol{m}, r_i) \\ &= w_{r_i} \cdot U_{0,w_{r_i}}(u_i)\, p(y_i \mid m_{r_i}) \\ &= I(u_i < w_{r_i})\, p(y_i \mid m_{r_i}). \end{aligned} \tag{3.4.5}$$

As a side note, it is possible to derive the distribution of $u_i \mid \boldsymbol{w}, \boldsymbol{m}$, which is

$$\begin{aligned} p(u_i \mid \boldsymbol{w}, \boldsymbol{m}) &= \int \sum_{h=1}^{\infty} w_h U_{0,w_h}(u_i)\, p(y_i \mid m_h)\, d\mathcal{L}(y_i) \\ &= \sum_{h=1}^{\infty} w_h U_{0,w_h}(u_i) \int p(y_i \mid m_h)\, d\mathcal{L}(y_i) \\ &= \sum_{h=1}^{\infty} w_h U_{0,w_h}(u_i) \\ &= \sum_{h=1}^{\infty} I(u_i < w_h). \end{aligned}$$

The latent variable u_i is therefore a mixture of uniform distributions.

The joint distribution from equation 3.4.5 can be used to derive the full conditionals

needed to set up the sampler:

1. simulate $v_h \mid \boldsymbol{r} \sim \text{Beta}\left(1 + n_h, \alpha + \sum_{l=h+1}^{r^\star} n_l\right)$, for $h = 1, \ldots, r^\star$;
2. simulate $u_i \mid \ldots \sim U\left(0, w_{r_i}\right),\ i = 1, \ldots, n$;
3. simulate $v_h \mid \boldsymbol{r} \sim \text{Beta}\left(1, \alpha\right)$, for $h = r^\star + 1, \ldots, h^\star$, where $h^\star$ is the smallest integer such that $\sum_{h=1}^{h^\star} w_h > 1 - \min\{u_1, \ldots, u_n\}$;
4. simulate $m_h \mid \ldots$ from $p\left(m_h \mid \boldsymbol{r}, \boldsymbol{y}\right) \propto G_0 \prod_{\{i:r_i=h\}} p\left(y_i \mid m_h\right)$, if cluster h is occupied, or from G_0 if cluster h is unoccupied, for $h = 1, \ldots, h^\star$;
5. simulate $r_i = h \mid \ldots$ from $p\left(r_i = h \mid \ldots\right) \propto I\left(u_i < w_h\right) p\left(y_i \mid m_h\right)$.

In step 1, for $h \leq r^\star$, where $r^\star$ is the maximum of $\{r_1, \ldots, r_n\}$, $v_h \mid \boldsymbol{r}$ can be simulated in the same fashion as equation 3.4.2:

$$v_h \mid \boldsymbol{r} \sim \text{Beta}\left(1 + n_h, \alpha + \sum_{l=h+1}^{r^\star} n_l\right). \tag{3.4.6}$$

Simulation of the first $h \leq r^\star$ components is sufficient to continue to step 2. When $h \leq r^\star$ corresponds to unoccupied clusters, v_h can be drawn from a Beta $(1, \alpha)$.

In step 2, Equation 3.4.5 can be rewritten as:

$$p\left(y_i, u_i, r_i \mid \boldsymbol{w}, \boldsymbol{m}\right) = w_{r_i} \cdot U_{0, w_{r_i}}\left(u_i\right) \cdot p\left(y_i \mid m_{r_i}\right), \tag{3.4.7}$$

which is a mixture whose components are driven by the value of r_i, and which is a shorthand for:

$$p\left(y_i, u_i, r_i = h \mid \boldsymbol{w}, \boldsymbol{m}\right) = \sum_{h=1}^{\infty} w_h \cdot U_{0, w_h}\left(u_i\right) \cdot p\left(y_i \mid m_h\right) \cdot \delta_h\left(r_i\right).$$

Further conditioning on r_i leads to:

$$p\left(y_i, u_i \mid r_i, \boldsymbol{w}, \boldsymbol{m}\right) = U_{0, w_{r_i}}\left(u_i\right) \cdot p\left(y_i \mid m_{r_i}\right), \tag{3.4.8}$$

and conditioning again on $y_i \mid r_i, \boldsymbol{w}, \boldsymbol{m}$ leads to

$$p\left(u_i \mid y_i, r_i, \boldsymbol{w}, \boldsymbol{m}\right) = U_{0, w_{r_i}}\left(u_i\right). \tag{3.4.9}$$

Step 3 is achieved by drawing further elements of v_h from a Beta $(1, \alpha)$. More elements need to be generated from $\boldsymbol{v} \mid \boldsymbol{r}$, to later support step 4 and allow sampling from $p(r_i = h \mid \ldots)$. Simulating from $p(r_i = h \mid \ldots)$ requires knowledge of the set $\{h : w_h > u_i\}$, which is finite; it is sufficient to find $\{h : w_h > u_i\}$, to sample $\boldsymbol{r}$. With respect to observation i, it is sufficient to find the smallest $h^\star$ such that

$$\sum_{h=1}^{h^\star} w_h > 1 - u_i,$$

where the left-hand side represents what is left of the stick in the stick-breaking process after node $h^\star$. To account for the fact that there are n observations, define

$$u^\star := \min\{u_1, \ldots, u_n\},$$

and modify the constraint above to:

$$u^\star > 1 - \sum_{h=1}^{h^\star} w_h.$$

In step 4, similarly to the finite DP Gibbs sampler, the full conditional of m_h is:

$$\begin{aligned} p(m_h \mid \boldsymbol{r}, \boldsymbol{y}, \boldsymbol{u}) &\propto p(\boldsymbol{y} \mid m_h, \boldsymbol{r}, \boldsymbol{u})\, p(m_h \mid \boldsymbol{r}, \boldsymbol{u}) \\ &\propto G_0(m_h) \prod_{\{i:r_i=h\}} p(y_i \mid m_h), \end{aligned}$$

which can be used to simulate from populated clusters. Unpopulated clusters may be simulated from G_0, if necessary.

In step 5, it is easy to see that equation 3.4.8 can be leveraged to obtain:

$$\begin{aligned} p(r_i = h \mid y_i, u_i, \boldsymbol{w}, \boldsymbol{m}) &\propto p(y_i, u_i \mid r_i, \boldsymbol{w}, \boldsymbol{m}) \cdot p(r_i \mid \boldsymbol{w}, \boldsymbol{m}) \\ &\propto U_{0,w_{r_i}}(u_i)\, p(y_i \mid m_{r_i}) \cdot w_{r_i} \\ &\propto \frac{1}{w_{r_i}} I(u_i < w_{r_i})\, p(y_i \mid m_{r_i})\, w_{r_i} \\ &\propto I(u_i < w_h)\, p(y_i \mid m_h). \end{aligned}$$

3.5 Inferring α

We now consider the case where α is random, with $\alpha \sim p(\alpha \mid \boldsymbol{\eta})$. Inference on α can be achieved by simply adding one more simulation step to the algorithms seen earlier in this section, where $\alpha \mid \ldots$ is updated conditional to all other parameters.

Escobar and West, 1995 observed that by sampling $\boldsymbol{\theta} \mid \boldsymbol{y}$, one also indirectly samples K_n, and that the full conditional of α only depends on $\boldsymbol{y}, \boldsymbol{\theta}$ through $K_n = k$:

$$\begin{aligned} p(\alpha \mid \boldsymbol{y}, \boldsymbol{\theta}, K_n = k) &= p(\alpha \mid K_n = k) \\ &\propto p(\alpha)\, p(K_n = k \mid \alpha). \end{aligned} \tag{3.5.1}$$

Recalling equation 2.1.12, equation 3.5.1 therefore can be rewritten as:

$$p(\alpha \mid K_n = k) \propto p(\alpha)\ \ \alpha^k\ \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)}, \tag{3.5.2}$$

which allows to draw from the conditional posterior of α.

In sections 3.5.1 and 3.5.2 we describe some practical steps for some distributional choices for α. Clearly, other choices are also possible, although they may require other simulation methods such as accept-reject or a Metropolis-Hastings step according to equation 3.5.1; we sketch the random walk Metropolis-Hastings algorithm in 3.5.3.

We conclude the topic of how to infer α with section 3.5.4, where we address the special case of the finite DP model, where an alternate approach to the one we outline above can be followed.

3.5.1 Gamma prior

This sampling algorithm was first introduced by Escobar and West, 1995, where it involved drawing $\eta \mid \alpha, K_n = k$ from a beta distribution, and then drawing $\alpha \mid \eta, K_n = k$ from a mixture of two gamma distributions. We present a simple variation of the same approach, resulting in a more parsimonious algorithm which as a second step only involves drawing from a gamma distribution rather than from

a mixture of gamma distributions, and which also appeared in Ghosal and Van der Vaart, 2017.

Recall from equation 3.5.2 that

$$p(\alpha \mid K_n = k) \propto p(\alpha)\ \alpha^k\ \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)}. \tag{3.5.3}$$

By the definition of the Γ and β functions, we have that

$$\frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} = \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)}\frac{\Gamma(n)}{\Gamma(n)} = \frac{\beta(\alpha,n)}{\Gamma(n)},$$

which allows to rewrite equation 3.5.3 as:

$$\begin{aligned} p(\alpha \mid K_n = k) &\propto p(\alpha)\,\alpha^k \frac{\beta(\alpha,n)}{\Gamma(n)} \\ &\propto p(\alpha)\,\alpha^k \beta(\alpha,n) && (3.5.4) \\ &\propto p(\alpha)\,\alpha^k \int_0^1 \eta^{\alpha-1}(1-\eta)^{n-1}\,\mathrm{d}\eta, && (3.5.5) \end{aligned}$$

Equation 3.5.5 can be interpreted as the marginal distribution obtained from the joint distribution of $(\alpha, \eta \mid K_n = k)$:

$$p(\alpha, \eta \mid K_n = k) \propto p(\alpha)\,\alpha^k \eta^{\alpha-1}(1-\eta)^{n-1},\ 0 < \eta < 1.$$

Hence, the conditional distribution of $\eta \mid \alpha, K_n = k$ is

$$p(\eta \mid \alpha, K_n = k) \propto \eta^{\alpha-1}(1-\eta)^{n-1},$$

and it is therefore a Beta(α, n). The conditional distribution of $\alpha \mid \eta, K_n = k$ is

$$p(\alpha \mid \eta, K_n = k) \propto p(\alpha)\,\alpha^k \eta^{\alpha-1},$$

which, when $p(\alpha)$ is a Gamma(a, b) density, leads to:

$$\begin{aligned} p(\alpha \mid \eta, K_n = k) &\propto \alpha^{a-1}\alpha^k e^{-b\alpha}\eta^{\alpha} \\ &\propto \alpha^{a+k-1} e^{-\alpha(b-\log\eta)}, \end{aligned}$$

which is proportional to the density of a Gamma$(a+k, b-\log\eta)$.

To summarise, when $\alpha \sim \text{Gamma}(a, b)$, the algorithm allows to sample $\alpha \mid K_n = k$ in two steps:

1. draw $\eta \mid \alpha, K_n = k$ from a beta distribution

$$\eta \mid \alpha, K_n = k \sim \text{Beta}(\alpha, n),$$

2. sample α from a gamma distribution

$$\alpha \mid \eta, K_n = k \sim \text{Gamma}(a + k, b - \log \eta).$$

It is easy to see that, when $p(\alpha)$ is a truncated Gamma instead (truncated at c and d respectively on the left and on the right), the only adjustment to the formula above is to sample α from a truncated Gamma at step 2 above.

3.5.2 Uniform prior

When $p(\alpha)$ is a Uniform (a, b) density, we have that:

$$\begin{aligned} p(\alpha \mid \eta, K_n = k) &\propto I_{[a,b]}(\alpha)\, \alpha^k \eta^{\alpha-1} (1-\eta)^{n-1} \\ &\propto I_{[a,b]}(\alpha)\, \alpha^k \eta^{\alpha} \\ &\propto I_{[a,b]}(\alpha)\, \alpha^k e^{\alpha \log \eta}, \end{aligned}$$

which can be easily recognised to be a truncated Gamma $(k+1, -\log \eta)$.

To summarise, when $\alpha \sim \mathcal{U}(a, b)$, the algorithm allows to sample $\alpha \mid K_n = k$ in two steps:

1. draw $\eta \mid \alpha, K_n = k$ from a beta distribution

$$\eta \mid \alpha, K_n = k \sim \text{Beta}(\alpha, n)$$

2. sample α from a truncated Gamma $(k+1, -\log \eta)$.

Note that a flat, improper uniform prior $\mathcal{U}(0, \infty)$ does not lead to a proper posterior, in that the integral of $\alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)}$ does not converge for $k = n-1$ nor $k = n$.

3.5.3 Random walk Metropolis-Hastings

Other choices of priors do not lend themselves to the analytic approach described at the beginning of this section, and a different approach is needed. A possible solution is to adopt a Metropolis-Hastings step within the Gibbs sampler. Due to its flexibility, a practical choice would be for example a random walk Metropolis-Hastings step, to sample from $p(\alpha \mid K_n = k)$.

A good reference for the theory behind the random walk Metropolis is Robert and Casella, 2004. In summary, random walk Metropolis-Hastings involves:

1. sampling $\alpha^\star := \alpha^{(b)} - \alpha^{(b-1)}$ from a symmetric distribution $p(\alpha^\star)$, such as for example a standard normal, a t distribution or a uniform distribution centred around 0;

2. setting

$$\alpha^{(b)} = \begin{cases} \alpha^{(b-1)} + \alpha^\star, & \text{with probability } \min\left(\dfrac{f\left(\alpha^{(b-1)} + \alpha^\star\right)}{f\left(\alpha^{(b-1)}\right)}, 1\right), \\ \alpha^{(b-1)}, & \text{otherwise.} \end{cases}$$

An element of difficulty of this algorithm is that the tuning parameter needs to be heuristically adjusted on a case-by-case basis.

3.5.4 Finite DP

In the stick-breaking framework, other than using the approach outlined at the beginning of section 3.5, one can also leverage the fact that $p(\alpha \mid \boldsymbol{w}, \boldsymbol{r}, \boldsymbol{m}, \boldsymbol{y})$ only depends on $\boldsymbol{w}$, and therefore its full conditional distribution is:

$$p(\alpha \mid \boldsymbol{r}, \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{y}) = p(\alpha \mid \boldsymbol{w}) \propto p(\alpha)\, p(\boldsymbol{w} \mid \alpha). \tag{3.5.6}$$

While in the exact stick-breaking framework $\boldsymbol{w}$ is infinite-dimensional, hence problematic, in the finite DP framework it is much easier to deal with (Ishwaran &

Zarepour, 2000): the probability distribution of $p(\boldsymbol{w} \mid \alpha)$ is easily obtained from equation 2.1.7, where all the terms that do not depend on α are unnecessary for our purpose. In the finite DP case, equation 3.5.6 can then be rewritten as

$$p(\alpha \mid \boldsymbol{w}) \propto p(\alpha)\, p(\boldsymbol{w} \mid \alpha) \propto p(\alpha)\, \alpha^{H-1} w_H^{\alpha-1}. \tag{3.5.7}$$

When $\alpha \sim \text{Gamma}(a_\alpha, b_\alpha)$, equation 3.5.7 can again be rewritten as

$$\begin{aligned} p(\alpha \mid \boldsymbol{w}) &\propto \alpha^{a_\alpha - 1} e^{-b_\alpha \alpha} \alpha^{H-1} w_H^{\alpha-1} \\ &\propto \alpha^{a_\alpha + H - 2} e^{-\alpha(b_\alpha - \log w_H)}, \end{aligned}$$

which, after accounting for the normalising constant, leads to :

$$\alpha \mid \boldsymbol{w} \sim \text{Gamma}\left(a_\alpha + H - 1, b_\alpha - \sum_{h=1}^{H-1} \log(1 - v_h)\right). \tag{3.5.8}$$

Although equation 3.5.8 is numerically more stable than the following alternative, it may be rewritten in a more compact form as:

$$\alpha \mid \boldsymbol{w} \sim \text{Gamma}(a_\alpha + H - 1, b_\alpha - \log w_H),$$

considering that $\sum_{h=1}^{H-1} \log(1 - v_h) = \log \prod_{h=1}^{H-1} (1 - v_h) = \log w_H$.

Chapter 4

Selection of the precision parameter hyperprior

The topic of this chapter is the Dirichlet process mixture model with random precision parameter α, inducing K_n clusters over n observations through its latent random partition. Our goal is to specify $p(\alpha \mid \boldsymbol{\eta})$, the prior distribution of α, in a way that is meaningful.

Previous studies have focused on the linkage between $p(\alpha \mid \boldsymbol{\eta})$ and $p(K_n)$ to draw conclusions on how to best choose $\boldsymbol{\eta}$ to reflect one's prior knowledge of $p(K_n)$. We show how such approaches, which all depend on the sample size n, rely on assumptions which break down asymptotically as $n \to \infty$, and while trying to be weakly informative about K_n, they end up being overly so on other important measures of interest. Common improper priors on the half-line offer no better outcome, as they lead to an improper posterior.

In this chapter, we assess the limitations of the aforementioned sample-size-dependent approaches to the choice of $\boldsymbol{\eta}$. We enrich their class by testing Jeffreys' prior in the DP framework, and by evaluating its behaviour. Finally, we propose an alternative methodology which does not depend on K_n or on the sample size, but rather on the relationship between the largest stick lengths in the stick-breaking construction, to informedly reflect one's prior belief onto $p(\alpha \mid \boldsymbol{\eta})$. We complete this chapter

with an example where existing sample-size-dependent approaches fail, while our sample-size-independent approach continues to be feasible.

4.1 Introduction

Typical usages of the DPM are density and cluster estimation; the former is motivated by the flexibility of the DPM, and the latter by the latent random partition that the model induces on the observed data. In both cases, the precision parameter α of a DPM is of great significance, since it influences the smoothness of the resulting density, as well as K_n, the random number of clusters in the underlying partition of n data points (Dorazio, 2009; Murugiah & Sweeting, 2012).

Methods have been developed to infer α from the data. For example, point estimates can be obtained with empirical Bayes, as outlined in Liu, 1996, or alternatively a full Bayes approach can be used to obtain its full posterior distribution. This chapter focuses on the latter, which often involves MCMC; we address the question of how to best choose the parameter vector $\boldsymbol{\eta}$ of the prior distribution $p(\alpha \mid \boldsymbol{\eta})$, and in particular what quantity of interest one's prior belief should be anchored to.

Existing approaches from statistical literature typically choose $\boldsymbol{\eta}$ based on how well it approximates one's prior belief of K_n; another common trait is that they try to induce a diffuse prior on K_n (Dorazio, 2009; Murugiah & Sweeting, 2012; West & Escobar, 1993). However, by leveraging K_n, they depend on the sample size, which is typically undesirable because the resulting modelling construct, including the data generating process, then becomes only applicable for one particular sample size: a prior deemed diffuse or weakly informative on K_n for a certain sample size n would not necessarily be so for a different value of n. In fact, articulating one's prior belief through K_n leads to a sequence of priors indexed by the sample size, $\{p(\alpha \mid \boldsymbol{\eta}_n)\}_{n=1}^{\infty}$. More importantly, we show in section 4.2 how, by trying to be diffuse in K_n, such priors actually influence some other important quantities of interest, in a way that is material. We also show how common improper priors such as $p(\alpha) \propto 1$ and

$p(\alpha) \propto (1/\alpha)$ do not offer a valid alternative either, as they lead to an improper posterior.

Our contribution is twofold. On the one hand, we enrich the class of sample-size-dependent priors by testing Jeffreys' prior in the DP framework, and by evaluating its behaviour. On the other hand, we then turn away from K_n, and we look at the stick-breaking representation of the underlying Dirichlet Process instead, to find an alternate sensible way of setting $\boldsymbol{\eta}$. In particular, we propose a new approach which is independent of n, and which leverages the infinite-dimensional collection of point masses induced by the Dirichlet Process, and the relationship between the relative size of the largest masses therein, to guide one's choice of $p(\alpha \mid \boldsymbol{\eta})$.

In the next sections, we proceed as follows. In section 4.2, we discuss the existing literature of α priors, including sample-size-dependent priors, improper priors, and their limitations. In section 4.3, we introduce new priors: first we enrich the class of sample-size dependent priors testing Jeffreys' prior in the DP framework, and by assessing its behaviour, and then we introduce a novel approach to the selection of $p(\alpha \mid \boldsymbol{\eta})$, which does not depend on the sample size. In section 4.4, we cross-examine how sample-size-dependent and sample-size-independent approaches perform in relation to each other. In section 4.5, we showcase an example where existing sample-size-dependent approaches fail, while our sample-size-independent approach continues to be feasible. Section 4.6 outlines our conclusions.

4.2 Existing priors for α

In this section we discuss two approaches to prior specification. Methods in the first group (see section 4.2.1) all leverage one's prior assumptions on K_n as a target to determine $\boldsymbol{\eta}$ in $p(\alpha \mid \boldsymbol{\eta})$; as such, they depend on the sample size n, and we refer to them as 'SSD' (sample-size-dependent). Those in the second group (see section 4.2.2 and 4.2.3) are quasi-degenerate and improper priors.

Although in principle any distributional choice of α is allowed, both approaches have historically been discussed by their authors in the context of $\alpha \sim \text{Ga}(a, b)$. The popularity of the gamma distribution as a prior for α is due to reasons of computational attractiveness of the posterior, and is to be traced back to Escobar and West, 1995 (see section 3.5.1).

4.2.1 Sample-size-dependent approaches

Methods in this group are the K_n-diffuse prior of West and Escobar, 1993, the DORO prior of Dorazio, 2009, and the SCAL prior of Murugiah and Sweeting, 2012.

K_n-diffuse prior

West and Escobar, 1993 use a gamma hyperprior, $\alpha \sim \text{Ga}(a, b)$, "supporting a diffuse range of reasonably large values consistent with possibly large values" of K_n. In their article, they use $n = 74$; their prior supports "a wide range of k values between about $k = 8$ and $k = 35$"[1]. We call their approach K_n-diffuse, to highlight that it is not necessarily diffuse in α, but rather it is diffuse in K_n.

We observe that this approach, which is appealing in its simplicity, has a dependency on the size of the sample it is applied to. For example, Figure 4.1 shows the impact on $p(K_n)$ of a $\text{Ga}(10, 1)$ prior, as n moves from $n = 10$ to $n = 100$:

- in the left panel ($n = 10$), K_n is centred on values that are large relative to n, with a wide spread relative to the support of K_n, hence it is well-diffused;
- in the right panel ($n = 100$), K_n is centred on smaller values (relative to n), with a smaller spread over the support of K_n – meaning that it is not as well-diffused as when $n = 10$.

[1] West and Escobar, 1993 use $a = 5$ and $b = 0.5$, and k in their notation is equivalent to K_n in ours.

As a result, if a K_n-diffuse prior is defined as one whose mass is well-spread around central values of K_n, the consequence is that, under a $\mathrm{Ga}(a, b)$ prior, (a, b) needs to be updated as n grows, to ensure that central values of K_n continue to be well covered, and that the relative spread around central values is preserved.

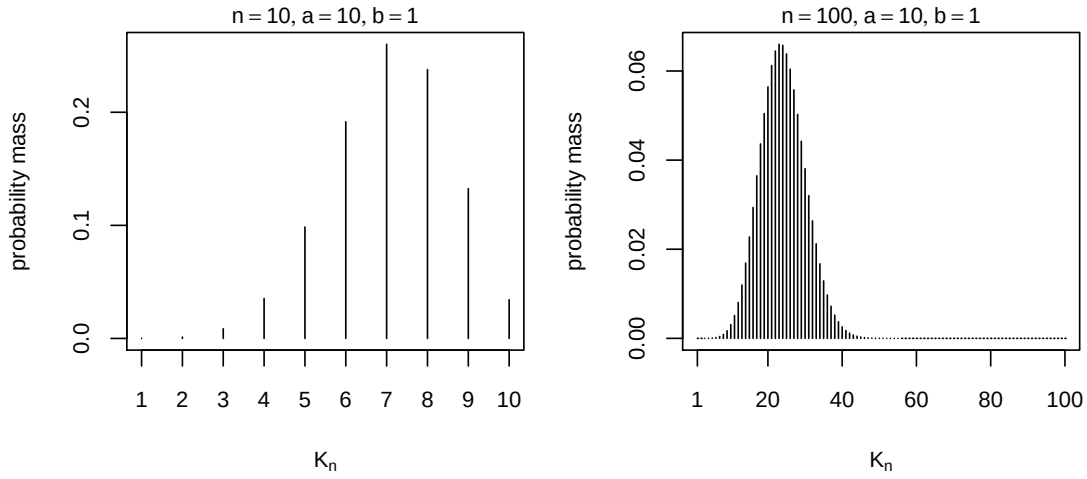

Figure 4.1: K_n-diffuse prior. While the prior probability distribution induced on K_n by $\alpha \sim \mathrm{Ga}(10, 1)$ appears reasonably diffuse for $n = 10$, it is less so for $n = 100$, as the shape and skewness of $p(K_n)$ change with n.

DORO priors

Dorazio, 2009 proposes $\alpha \sim \mathrm{Ga}(a, b)$, with (a, b) set to minimise the Kullback-Leibler distance between the prior probability distribution $p(K_n)$ induced by $p(\alpha \mid \boldsymbol{\eta})$ on K_n, and a target discrete distribution; in the absence of prior information about K_n, Dorazio, 2009 uses the discrete uniform as a target. The DORO approach results in a pre-determined list of optimal values of (a, b) for various values of n (Table 4.1).

However, Figure 4.2 shows that the approximation of the discrete uniform resulting from the DORO prior is visibly coarse, and that it does not appear to improve as n grows. In fact, asymptotic results show that, when $\alpha \sim \mathrm{Ga}(a, b)$, K_n converges sub-linearly to a negative binomial random variable, meaning that a discrete uniform is unachievable for $n \to \infty$: from equation 2.1.14,

$$p(K_n \mid \alpha) \xrightarrow{d} \mathrm{Poi}(\alpha \log n), \quad n \to \infty,$$

and $\alpha \log n \sim \mathrm{Ga}\,(a, b/\log n)$ hence $K_n \sim \mathrm{NB}\,(a, b/\,(b + \log n))$ in the limit.

Furthermore, as we will see in section 4.3.1, the shape of the prior induced by DORO on K_n is also quite different from the one that is attained under Jeffreys' prior, which is one more reason why the choice by DORO of targeting a discrete uniform prior distribution on K_n can be debated.

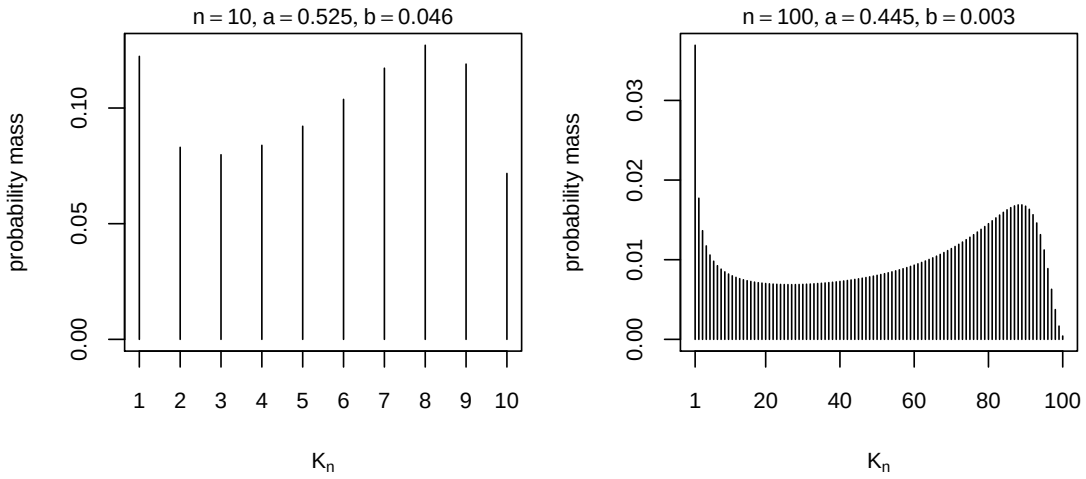

Figure 4.2: DORO prior. Prior probability distribution induced on K_n by $\alpha \sim \mathrm{Ga}\,(a, b)$. K_n does not appear to be close to the target discrete uniform distribution, and the approximation does not appear to improve as n increases.

n	a	b	D_{KL}
5	0.541	0.096	0.00458
10	0.525	0.046	0.01904
15	0.512	0.029	0.03048
20	0.501	0.021	0.03942
25	0.490	0.015	0.04660
30	0.486	0.013	0.05272
35	0.480	0.010	0.05806
40	0.475	0.009	0.06265
45	0.470	0.008	0.06684
50	0.467	0.007	0.07050
100	0.445	0.003	0.09529

Table 4.1: Optimal values of a, b under the DORO approach, when $\alpha \sim \mathrm{Ga}\,(a, b)$ and when the target distribution $p\,(K_n)$ is the discrete uniform. We have enriched the original table from Dorazio, 2009 with an additional entry for $n = 100$.

SCAL priors

Murugiah and Sweeting, 2012 propose a scaling approach, where the values of (a, b) are initially computed for a given n according to how well they perform at recovering some known cluster structure; (a, b) are subsequently rescaled to other choices of n, without the need to re-compute them again through the fully-fledged determination process.

Upon the initial determination of (a, b), the goal of SCAL is to scale (a, b) in such a way that the prior mean of K_n is only affected to a small extent by the changes, while the variance is influenced to a larger extent. In particular, SCAL is based on fixing $p(K_n = 1)$ and $p(K_n \in \{c, c+1, \ldots, n-1, n\})$ to some determined value, for a suitably selected c, and deriving (a, b) implicitly as n varies, with equation 2.1.13. Murugiah and Sweeting, 2012 suggest $c = \lceil c_0 \log n \rceil$, with $c_0 = 2$.

Murugiah and Sweeting, 2012 use simulated data sets of $n = 6$ observations to elicit $(a, b) = (1, 1)$; they then test their scaling approach up to $n = 25$, by re-calculating (a, b) while keeping $p(K_n = 1) = 0.34$ and $p(K_n \in \{c, \ldots, n\}) = 0.15$. They fit a curve through the results that they obtain, to ultimately propose the following equation, as an easier approximation[2] to SCAL:

$$a = b = e^{-0.033n}. \tag{4.2.1}$$

Although Murugiah and Sweeting, 2012 only obtained the optimal (a, b) values for their test case for $n \in \{6, 10, 15, 20, 25\}$, we extend their results to $n \in \{50, 75, 100\}$ (see table 4.2). As expected, we observe dependence on n, albeit to a lesser extent than DORO. We also note that the proposed approximation significantly diverges from the exact values for $n > 25$; in fact, instead of the exact values $(a, b) = (0.403, 0.370)$, for $n = 100$ it yields $a = b = 0.037$, a quasi-degenerate gamma prior.

[2]This is justified in Murugiah and Sweeting, 2012 by observing that $\mathbb{E}[\alpha] = 1$, which is fixed, and that $\mathbb{V}[\alpha] = e^{0.0165n}$, which increases with n, as their approach originally intended.

n	exact (a, b)	approx. (a, b)
25	$(0.490, 0.438)$	$(0.438, 0.438)$
50	$(0.466, 0.467)$	$(0.192, 0.192)$
75	$(0.432, 0.420)$	$(0.084, 0.084)$
100	$(0.403, 0.370)$	$(0.037, 0.037)$

Table 4.2: Optimal and approximate values of (a, b) under the SCAL approach, as we determined them to be according to the approach outlined in Murugiah and Sweeting, 2012. Approximate values originate from equation 4.2.1. Targets are $p(K_n = 1) = 0.34$ and $p(K_n \in \{c, \ldots, n\}) = 0.15$, $c = c_0 \log n$, $c_0 = 2$.

4.2.2 Quasi-degenerate priors

Similarly to the SCAL approximation above, other authors have independently suggested, on isolated occasions rather than as part of a prior elicitation framework, to use a quasi-degenerate $\mathrm{Ga}(a, b)$, with a and b close to zero, on the basis that it approximates the improper prior $p(\alpha) \propto 1/\alpha$ that is uniform in $\log(\alpha)$ (Escobar & West, 1995, 1998; Lunn et al., 2012; Navarro et al., 2006).

However, gamma priors with very small (a, b) are undesirable, because

$$\lim_{(a,b)\to(0,0)} p(K_n = 1) = 1,$$

hence they lead to a prior expectation of a parametric model with only one mixture component, which negates the reason for using a nonparametric prior in the first place and which is likely to overwhelm the data due to its strength; Dorazio, 2009 and Murugiah and Sweeting, 2012 also draw similar conclusions.

This is proven as follows. We are interested in

$$\lim_{(a,b)\to(0,0)} p(K_n = k) = s_{n,k} \lim_{(a,b)\to(0,0)} \int_0^\infty \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \mathrm{d}p(\alpha \mid a, b), \tag{4.2.2}$$

where $p(\alpha \mid a, b)$ is the gamma probability measure with parameters (a, b). In what follows, we re-write equation 4.2.2 as the limit of a sequence, and we use results on the weak convergence of measures to prove that the sequence converges to 1 for $k = 1$, and to 0 for every other admissible value of k.

For each path in $\mathbb{R}^+ \times \mathbb{R}^+$ where $(a, b) \to (0, 0)$, we define $\{X_l\}$, a sequence of gamma distributed random variables $X_1, X_2, \ldots$, each with parameters $(a_1, b_1), (a_2, b_2), \ldots$ identified along that path. The distribution function of X_l is

$$F_{X_l}(\alpha; a_l, b_l) = \frac{\gamma(a_l, b_l\alpha)}{\Gamma(a_l)},$$

where $\gamma()$ is the lower incomplete gamma function. It is well known that

$$\frac{\gamma(s, t)}{t^s} \to \frac{1}{s},$$

as $t \to 0$. Hence

$$\begin{aligned}\lim_{(a,b)\to(0,0)} \gamma(a, b\alpha)\frac{1}{\Gamma(a)} &= \lim_{(a,b)\to(0,0)} \frac{(b\alpha)^a}{a}\frac{1}{\Gamma(a)} = \lim_{l\to\infty} \frac{(b_l\alpha)^{a_l}}{a_l}\frac{1}{\Gamma(a_l)} \\ &= \lim_{l\to\infty} (b_l\alpha)^{a_l}\frac{1}{a_l\Gamma(a_l)} = \lim_{l\to\infty} {b_l}^{a_l} \cdot \lim_{l\to\infty} \alpha^{a_l} = 1, \quad \forall\alpha > 0,\end{aligned}$$

where we use the fact that

$$\lim_{(a,b)\to(0,0)} b^a = \lim_{r\to 0} (r\sin\theta)^{r\cos\theta} = \lim_{r\to 0} (r^r)^{\cos\theta} (\sin^r\theta)^{\cos\theta} = 1,$$

which holds as the paths along $a = 0$ and $b = 0$ do not belong to the function domain $\mathbb{R}^+ \times \mathbb{R}^+$ of F_{X_l}.

Therefore $X_l \xrightarrow{d} 0$, since the distribution function of the r.v. $X = 0$ is equal to 1 over the continuity set $(0, \infty)$ of X, for every path and every sequence $\{X_l\}$. A consequence is that the sequence of gamma probability measures $\{p_l\}$ induced by $\{X_l\}$ converges weakly to the Dirac measure δ_0. Hence equation 4.2.2 becomes

$$\lim_{(a,b)\to(0,0)} p(K_n = k) = s_{n,k}\int_0^\infty \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \mathrm{d}\delta_0(\alpha) = \begin{cases} 1, & k = 1, \\ 0, & k = 2, \ldots, n, \end{cases}$$

as $\alpha^k\Gamma(\alpha)/\Gamma(\alpha + n)$ is bounded and continuous.

4.2.3 Improper priors

In the preceding chapter, we mentioned how some studies motivate the use of quasi-degenerate gamma priors on the basis that they approximate the improper prior $p(\alpha) \propto 1/\alpha$.

However, using $p(\alpha) \propto 1/\alpha$ leads to an improper posterior $p(\alpha \mid \boldsymbol{y})$ as well as to an improper implied prior $p(K_n)$, and so does $p(\alpha) \propto 1$ too. In fact, from equation 2.1.12 we obtain

$$p(\alpha \mid K_n = k) \propto \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \cdot \frac{1}{\alpha} \sim \frac{1}{\alpha^{n-k+1}} \quad \text{as } \alpha \to \infty,$$

which is not integrable on any (t, ∞) interval in the domain for $k = n$, meaning that the α posterior induced by the prior $1/\alpha$ cannot be normalised and is improper.

The $p(\alpha) \propto 1/\alpha$ prior also induces an improper prior on K_n, as

$$p(K_n = k) = s_{n,k} \int_0^\infty \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \cdot \frac{1}{\alpha} \mathrm{d}\alpha,$$

where the integrand diverges as above.

Similar considerations apply when the α prior is $p(\alpha) \propto 1$, the conclusion being that the α posterior and the induced K_n prior are improper because they are divergent for $k = n$ and $k = n - 1$.

4.3 New priors for α

In this section we make two main contributions. One is to enrich the class of SSD priors by testing Jeffreys' prior in the DP framework, and by evaluating its behaviour. The other is to propose a method which does not depend on the sample size, but rather on the relationship between the largest stick lengths in the stick-breaking construction, to informedly reflect one's prior belief onto $p(\alpha \mid \eta)$.

4.3.1 Jeffreys' prior

This prior[3] was initially introduced in Jeffreys, 1946 as one that is invariant under a reparameterisation $p_\zeta(\zeta)$ of $p_\alpha(\alpha)$:

$$p_\zeta(\zeta) = p_\alpha(\alpha)\left|\frac{\mathrm{d}\alpha}{\mathrm{d}\zeta}\right|.$$

Subsequently, it was found to have other desirable properties too, and it is now widely used in statistics and best known as the most prominent example of an *objective* prior. For univariate cases like ours, it is obtained as:

$$p_\alpha(\alpha) \propto \sqrt{I_\alpha(\alpha)},$$

where $I_\alpha(\alpha)$ is the Fisher information:

$$\begin{aligned} I_\alpha(\alpha) &= -\mathbb{E}_{\boldsymbol{y}}\left[\frac{\partial^2}{\partial\alpha^2}\log p(\boldsymbol{y} \mid \boldsymbol{\alpha})\right] \\ &= \mathbb{E}_{\boldsymbol{y}}\left[\left(\frac{\partial}{\partial\alpha}\log p(\boldsymbol{y} \mid \alpha)\right)^2\right] \\ &= \mathbb{E}_k\left[\left(\frac{\partial}{\partial\alpha}\log p(k \mid \alpha)\right)^2\right], \end{aligned}$$

and the last passage is justified by the fact that, in the DPM model, k is a sufficient statistic for α, hence the equality.

Its calculation yields:

$$\begin{aligned} p(\alpha) \propto \sqrt{I_\alpha(\alpha)} &= \sqrt{\mathbb{E}_k\left[\left(\frac{\partial}{\partial\alpha}\log p(k \mid \alpha)\right)^2\right]} \\ &= \sqrt{\mathbb{E}_k\left[\left(\frac{\partial}{\partial\alpha}\left(\log s_{n,k} + k\log\alpha + \log\Gamma(\alpha) - \log\Gamma(\alpha+n)\right)\right)^2\right]} \\ &= \sqrt{\mathbb{E}_k\left[\left(\frac{k}{\alpha} + \psi_0(\alpha) - \psi_0(\alpha+n)\right)^2\right]} \\ &= \sqrt{\mathbb{E}_k\left[\frac{k^2}{\alpha^2} + 2\frac{k}{\alpha}\left(\psi_0(\alpha) - \psi_0(\alpha+n)\right) + \left(\psi_0(\alpha) - \psi_0(\alpha+n)\right)^2\right]} \end{aligned}$$

[3] I thank my Supervisor for the idea of exploring Jeffreys' prior, and for his contribution to the derivation of its formula.

$$
\begin{aligned}
&= \left\{ \frac{1}{\alpha^2} \left[\sum_{i=1}^{n} \left(\frac{\alpha (i-1)}{(\alpha+i-1)^2} \right) + \alpha^2 \left(\sum_{i=1}^{n} \frac{1}{\alpha+i-1} \right)^2 \right] + \right. \\
&\quad \left. + 2 \left(\sum_{i=1}^{n} \frac{1}{\alpha+i-1} \right) (\psi_0(\alpha) - \psi_0(\alpha+n)) + (\psi_0(\alpha) - \psi_0(\alpha+n))^2 \right\}^{0.5} \\
&= \left\{ \left(\frac{1}{\alpha} \sum_{i=1}^{n} \frac{i-1}{(\alpha+i-1)^2} \right) + \left(\sum_{i=1}^{n} \frac{1}{\alpha+i-1} \right)^2 - 2 \left(\sum_{i=1}^{n} \frac{1}{\alpha+i-1} \right)^2 + \right. \\
&\quad \left. + (\psi_0(\alpha) - \psi_0(\alpha+n))^2 \right\}^{0.5} \\
&= \sqrt{\frac{1}{\alpha} \sum_{i=1}^{n} \frac{i-1}{(\alpha+i-1)^2}} = \frac{\sqrt{\mathbb{V}\mathrm{ar}(k)}}{\alpha},
\end{aligned}
$$

which is justified by equation 2.1.11, and by the fact that, by definition, $\psi_0(z+1) = \psi_0(z) + \frac{1}{z}$ hence

$$
\psi_0(\alpha) - \psi_0(\alpha+n) = -\left(\frac{1}{\alpha} + \ldots + \frac{1}{\alpha+n-1} \right) = -\sum_{i=1}^{n} \frac{1}{\alpha+i-1}.
$$

As $\psi_1(\alpha+1) = \psi_1(\alpha) - \frac{1}{\alpha^2}$, we may also equivalently write the following, which we found to be computationally faster:

$$
p(\alpha) \propto \sqrt{\frac{1}{\alpha} \left[\psi_0(\alpha+n) - \psi_0(\alpha+1) + \alpha (\psi_1(\alpha+n) - \psi_1(\alpha+1)) \right]}.
$$

Propriety of Jeffreys' prior

We now attempt to determine if the so obtained prior is proper, whether it induces a proper posterior, and what its properties are. First, we observe that its density is 0 on the entire half-line for $n = 1$, and its integral also is, hence this prior carries no meaning for $n = 1$. We then move on to consider the more general case when $n > 1$. We use the comparison theorem to show that $f(\alpha) \leq g(\alpha)$ and therefore if $g(\alpha)$ is integrable then $f(\alpha)$ also is.

Since $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for any two $a, b > 0$, we have

$$
\sqrt{\frac{1}{\alpha} \sum_{i=2}^{n} \frac{i-1}{(\alpha+i-1)^2}} \leq \sqrt{\frac{1}{\alpha}} \left(\sqrt{\sum_{i=2}^{n-1} \frac{i-1}{(\alpha+i-1)^2}} + \sqrt{\frac{n-1}{(\alpha+n-1)^2}} \right),
$$

which by recursive application and by taking the integral leads to

$$\int_0^\infty \sqrt{\frac{1}{\alpha}\sum_{i=2}^{n}\frac{i-1}{(\alpha+i-1)^2}}\mathrm{d}\alpha \leq \sum_{i=2}^{n}\left(\int_0^\infty \sqrt{\frac{1}{\alpha}\frac{i-1}{(\alpha+i-1)^2}}\mathrm{d}\alpha\right) = (n-1)\,\pi < \infty,$$

as, for $i = 2, 3, \ldots$, the inner integral is

$$\int_0^\infty \sqrt{\frac{1}{\alpha}\frac{i-1}{(\alpha+i-1)^2}}\mathrm{d}\alpha = \pi.$$

Jeffreys' prior is therefore proper.

For $n = 2$, Jeffreys' prior can be expressed analytically. Its unnormalised cumulative probability function is:

$$p\,(\alpha \leq x) = \int_0^x \sqrt{\frac{1}{\alpha}\frac{1}{(\alpha+1)^2}}\mathrm{d}\alpha = 2\mathrm{atan}\sqrt{x},$$

therefore its normalising constant is the inverse of

$$\lim_{x\to\infty} 2\mathrm{atan}\sqrt{x} = \pi,$$

leading to the density and cumulative probability distribution functions:

$$p\,(\alpha) = \frac{1}{\pi\,(\alpha+1)\,\sqrt{\alpha}}, \tag{4.3.1}$$

$$p\,(\alpha \leq x) = \frac{2}{\pi}\mathrm{atan}\sqrt{\alpha}. \tag{4.3.2}$$

We do not have an analytical expression for $n > 2$. We have used numerical integration to obtain the normalising constant for $n = 10$ and $n = 100$, to produce Figure 4.3, which exemplifies the change in shape of $p\,(\alpha)$ as n increases.

Properties of Jeffreys' prior

Jeffreys' prior has no finite mean (and therefore it has no finite higher moments either):

$$\begin{aligned}\int_0^\infty \alpha p\,(\alpha)\,\mathrm{d}\alpha &= \int_0^\infty \alpha\cdot\frac{1}{\sqrt{\alpha}}\sqrt{\sum_{i=1}^{n}\frac{i-1}{(\alpha+i-1)^2}}\mathrm{d}\alpha \\ &> \sqrt{\sum_{i=2}^{n} i-1}\int_0^\infty \frac{\sqrt{\alpha}}{\alpha+i-1}\mathrm{d}\alpha =\end{aligned}$$

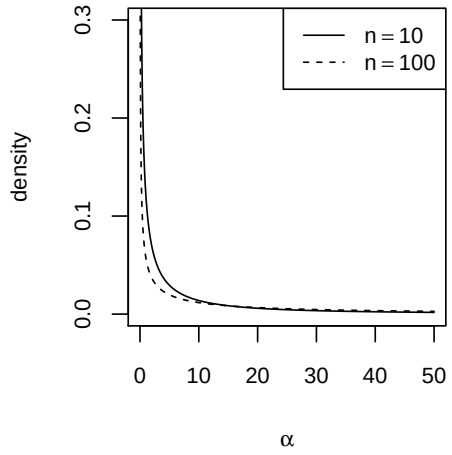

Figure 4.3: Density of Jeffreys' prior for $n = 10$ and $n = 100$.

$$= \sqrt{\sum_{i=2}^{n} i - 1} \cdot \left| \frac{2\,(1-n)\,\operatorname{atan}\sqrt{\frac{\alpha}{n-1}}}{\sqrt{n-1}} + 2\sqrt{\alpha} \right|_0^{\infty} = \infty.$$

In fact, if we take for example $n = 2$ and we attempt to simulate from Jeffreys' prior with the approach that we described in section 4.3.1, we obtain a chart where the sample mean does not appear to converge to any value.

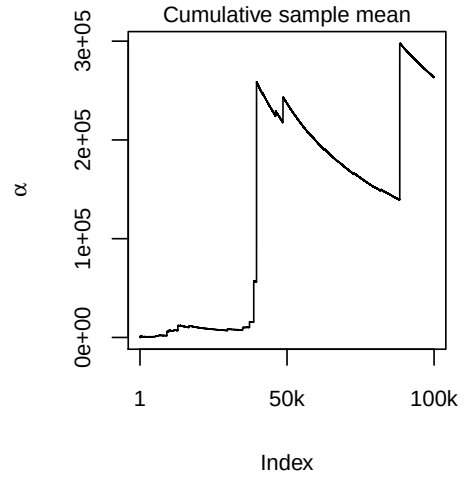

Figure 4.4: Cumulative sample mean of Jeffreys' prior for $n = 2$, over 100 thousand draws. It does not appear to converge to any value, as this distribution has no finite mean.

Propriety of the posterior induced by Jeffreys' prior on α

Jeffreys' prior induces a proper posterior $p\,(\alpha \mid K_n = k)$. Given

$$p\,(\alpha \mid K_n = k) \propto p\,(K_n = k \mid \alpha)\,p\,(\alpha)$$

$$\propto \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \sqrt{\frac{1}{\alpha} \sum_{i=1}^{n} \frac{i-1}{(\alpha+i-1)^2}},$$

its limit for $\alpha \to 0$ is

$$\lim_{\alpha \to 0} p(\alpha \mid K_n = k) = \alpha^{k-1} \frac{\Gamma(1)}{\Gamma(n)} \alpha^{-\frac{1}{2}} \sqrt{\sum_{i=1}^{n} \frac{1}{i-1}} \approx \alpha^{k-\frac{3}{2}},$$

which is convergent over $(0,1)$ (limit comparison test). Its limit for $\alpha \to \infty$ is

$$\lim_{\alpha \to \infty} p(\alpha \mid K_n = k) = \alpha^{k-n-\frac{3}{2}},$$

which is proper over (t, ∞), for $t \geq 1$.

Propriety of the prior induced by Jeffreys' prior on K_n

Recall from equation 2.1.12 the expression for $p(K_n = k \mid \alpha)$. The prior it induces on K_n is

$$\begin{aligned} p(K_n = k) &= s_{n,k} \int_0^\infty \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \sqrt{\frac{1}{\alpha} \sum_{i=1}^{n} \frac{i-1}{(\alpha+i-1)^2}} \mathrm{d}\alpha \\ &\propto s_{n,k} \int_0^\infty \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \alpha^{k-\frac{1}{2}} \sqrt{\sum_{i=1}^{n} \frac{i-1}{(\alpha+i-1)^2}} \mathrm{d}\alpha. \end{aligned}$$

Denoting the integrand by $g(\alpha)$,

$$\begin{aligned} \lim_{\alpha \to 0} g(\alpha) &= \lim_{\alpha \to 0} \frac{\Gamma(\alpha+1)}{\Gamma(\alpha+n)} \alpha^{k-\frac{3}{2}} \sqrt{\sum_{i=1}^{n} \frac{i-1}{(\alpha+i-1)^2}} \\ &= \lim_{\alpha \to 0} \frac{\alpha^{k-\frac{3}{2}}}{\Gamma(n)} \sqrt{\sum_{i=1}^{n} \frac{1}{i-1}} \\ &\propto \alpha^{k-\frac{3}{2}}, \end{aligned}$$

which does converge over $(0,1)$, and

$$\lim_{\alpha \to \infty} g(\alpha) \propto \alpha^{k-\frac{1}{2}-n-1},$$

which also does converge over $(1, \infty)$. Two examples of the shape of the prior distribution induced on K_n by assigning Jeffreys' prior to α are plotted in figure 4.5. This distribution appears to be approximately symmetric around $n/2$, for various

values of n that we have tried (we have tried up to $n = 1000$).

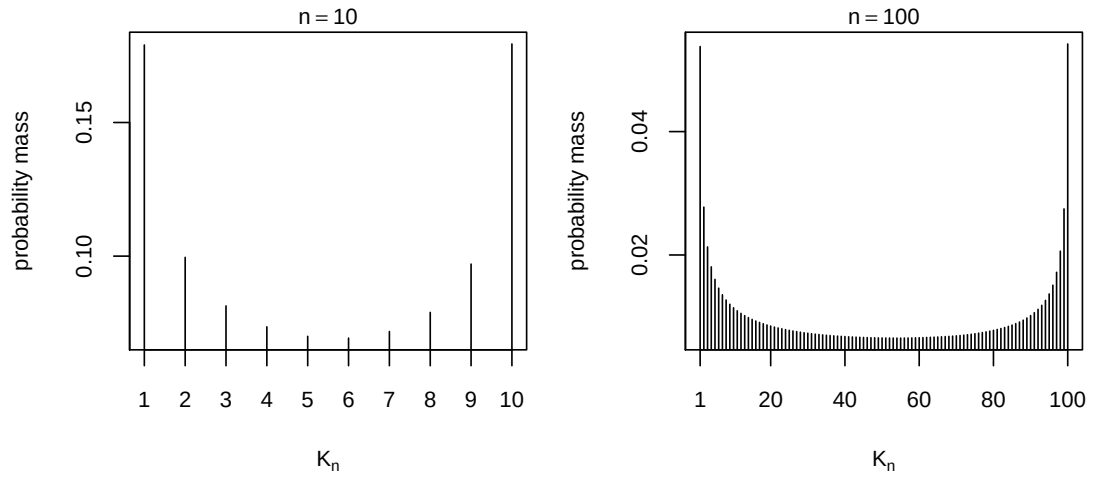

Figure 4.5: Prior distribution $p(K_n)$ induced by assigning Jeffreys' prior to $p(\alpha)$, for $n = 10$ and $n = 100$.

Sampling from Jeffreys' prior

As Jeffreys' prior for the Dirichlet process is a non-standard distribution, there is value in discussing how to sample from it. Like any other distribution which there are no widely known sampling methods for, one can resort to any of the general-purpose approaches described, for example, in Robert and Casella, 2004, such as the accept-reject method, importance sampling, the 2-d slice sampler or Metropolis-Hastings algorithms.

The accept-reject algorithm requires that a constant M exists such that the target distribution f and the proposal distribution g meet the condition $\frac{f(\alpha)}{g(\alpha)} \leq M$, for all values of α. This can be achieved by using Jeffreys' prior for $n = 2$ (which, as we have shown above, has an analytical solution) as the proposal distribution g; easy calculations lead then to

$$\frac{f(\alpha)}{g(\alpha)} = \sqrt{1 + 2\frac{(\alpha+1)^2}{(\alpha+2)^2} + \ldots + (n-1)\frac{(\alpha+1)^2}{(\alpha+n-1)^2}} \leq M = \sqrt{\sum_{i=1}^{n-1} i}.$$

Once M is determined, the steps to the algorithm are:

- generate X from the proposal distribution g, and generate $U \sim \text{Unif}(0, 1)$;

- accept the value above if $U \leq f(X) / (Mg(X))$; return to the step above if otherwise.

We also implemented, to compare:

- the 2d slice sampler,
- the independence Metropolis-Hastings algorithm with a Jeffreys' prior distribution ($n = 2$),
- the random-walk Metropolis algorithm with a half-Cauchy proposal and with a normal proposal distribution,

and found their convergence speed to be fastest for the 2d slice sampler, followed by the independence Metropolis-Hastings algorithm with Jeffreys' prior distribution and $n = 2$, followed again by the random-walk Metropolis with half-Cauchy proposal.

4.3.2 Sample-size-independent priors for α (SSI)

We introduce a new prior selection approach which is independent of sample size, and which is motivated by the fact that, irrespective of how many K_n clusters are observed in a sample of size n, there is always an underlying infinite-dimensional collection of point masses induced by the DP (α, G_0), which is independent of n and which one's prior belief can be reflected on. In particular, our approach is based on the size-biased random permutation of the infinite point masses, or alternatively on their ranked permutation. We henceforth refer to this approach as SSI (sample-size-independent).

Size-biased weights

In the stick-breaking representation of equation 2.1.5, a DP is a collection of infinitely many point masses. In particular, $\{w_h\}_{h=1}^{\infty}$ is a sequence of stochastically decreasing

weights; it is said to be a size-biased random permutation of the weights because the probability of each weight w_i being in first position in the size-biased permutation is precisely w_i (see section 2.1.2 and Pitman, 1996a). A priori, the probability distribution of the weights as they naturally arise in the stick-breaking construction of equation 2.1.4 and 2.1.5 is the same as the probability distribution of the size-biased weights, hence for simplicity we do not distinguish between them in notation, in this chapter, although in later chapters we write $\tilde{w}_i$ to specifically tell the size-biased weights apart from the weights in the order whereby they arise in the stick-breaking construction.

The conditional joint probability distribution of the first H size-biased weights is a particular case of the generalised Dirichlet distribution (Connor & Mosimann, 1969):

$$p\left(w_1, \ldots, w_H \mid \alpha\right) = \frac{\alpha^H}{\left(1-w_1\right)\ldots\left(1-w_1-\ldots-w_{H-1}\right)} \left(1-w_1-\ldots-w_H\right)^{\alpha-1}. \tag{4.3.3}$$

We plot its first two elements in Figure 4.6, for selected values of α. Since $p\left(w_1, w_2, \ldots\right)$ is infinite-dimensional, we restrict our analysis to a finite number of dimensions; for practical purposes and for simplicity, we focus on the bivariate distribution of w_1 and w_2; our approach can in principle be extended to more dimensions, if necessary. Analytically, we have that:

$$p\left(w_1, w_2 \mid \alpha\right) = \frac{\alpha^2}{1-w_1} \left(1-w_1-w_2\right)^{\alpha-1}.$$

The density $p\left(w_1, w_2 \mid \alpha\right)$ attains its maximum on $w_2 = 1 - w_1$ for $\alpha < 1$, at $(1, 0)$ for $1 \leq \alpha < 2$, at $w_2 = 0$ for $\alpha = 2$ and at $(0, 0)$ for $\alpha > 2$ (Figure 4.6). Its partial derivatives are:

$$\begin{aligned}
\frac{\partial p\left(w_1, w_2 \mid \alpha\right)}{\partial w_1} &= \alpha^2 \frac{\left(1-w_1-w_2\right)^{\alpha-2}}{\left(1-w_1\right)^2} \left(\left(\alpha-2\right) w_1 - w_2 - \alpha + 2\right), \\
\frac{\partial p\left(w_1, w_2 \mid \alpha\right)}{\partial w_2} &= \alpha^2 \left(1-\alpha\right) \frac{\left(1-w_1-w_2\right)^{\alpha-2}}{1-w_1},
\end{aligned}$$

and analysis of their sign leads to the considerations in Table 4.3.

Our prior belief in the cases displayed in Table 4.3 can be used to inform our choice

α	$p(w_1 \mid w_2, \alpha)$	$p(w_2 \mid w_1, \alpha)$
$0 < \alpha < 1$	increasing	increasing
$\alpha = 1$	increasing	constant
$1 < \alpha < 2$	concave; max. attained at $w_1 = 1 - \frac{w_2}{2-\alpha}$	decreasing
$\alpha = 2, w_2 \neq 0$	decreasing	decreasing
$\alpha = 2, w_2 = 0$	constant	decreasing
$\alpha > 2$	decreasing	decreasing

Table 4.3: Sample-size-independent approach, size-biased. Behaviour of $w_1 \mid w_2, \alpha$ and $w_2 \mid w_1, \alpha$, for different values of α.

of the parameter vector $\boldsymbol{\eta}$ of the prior distribution $p(\alpha \mid \boldsymbol{\eta})$, since assigning a prior to $p(\alpha \mid \boldsymbol{\eta})$ means mixing over those cases.

Denote by $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_d)$ the d parameters of the continuous probability distribution $p(\alpha \mid \boldsymbol{\eta})$, and denote by $p_{0,t_1}, \ldots, p_{t_d,\infty}$ our prior belief associated with a partition of $(0, \infty)$ into $d+1$ nonempty subsets $\{(0, t_1], \ldots, (t_d, \infty)\}$. We choose $\boldsymbol{\eta}$ so that $p(\alpha \mid \boldsymbol{\eta})$ reflects our prior belief by solving:

$$\begin{cases} p(0 < \alpha \leq t_1 \mid \boldsymbol{\eta}) = p_{0,t_1}, \\ p(t_1 < \alpha \leq t_2 \mid \boldsymbol{\eta}) = p_{t_1,t_2}, \\ \ldots \\ p(\alpha > t_d \mid \boldsymbol{\eta}) = p_{t_d,\infty}. \end{cases}$$

For example, when $d = 2,\ t_1 = 1,\ t_2 = 2$, we partition $(0, \infty)$ into $(0, 1]$,$(1, 2]$ and $(2, \infty)$.

The resulting system of equations can be either solved analytically, if the cumulative probability distribution admits an explicit representation of its inverse, or numerically. For example, this is analytically feasible when $\alpha \sim \text{Exp}(\eta)$, and we obtain:

$$\eta = \log\left((1 - p_{0,t_1})^{-\frac{1}{t_1}}\right), \tag{4.3.4}$$

and clearly $d = 1$, $p_{t_2,\infty} = 1 - p_{0,t_1}$.

When instead $\alpha \sim \text{Ga}(\eta_1, \eta_2)$, we obtain:

$$\begin{cases} \dfrac{\gamma(\eta_1, \eta_2 t_1)}{\Gamma(\eta_1)} = p_{0,t_1}, \\ \dfrac{\gamma(\eta_1, \eta_2 t_2) - \gamma(\eta_1, \eta_2 t_1)}{\Gamma(\eta_1)} = p_{t_1,t_2}. \end{cases}$$

For values of t_1, t_2 that mirror those from table 4.3, and for some arbitrary choices of the underlying probabilities, we obtain the results in table 4.4. These results are purely for exemplification, and the approach that we outline in this section can be used to calculate $\boldsymbol{\eta}$ for any partition of $(0, \infty)$, any associated probabilities, and any distributional choice of $p(\alpha \mid \boldsymbol{\eta})$.

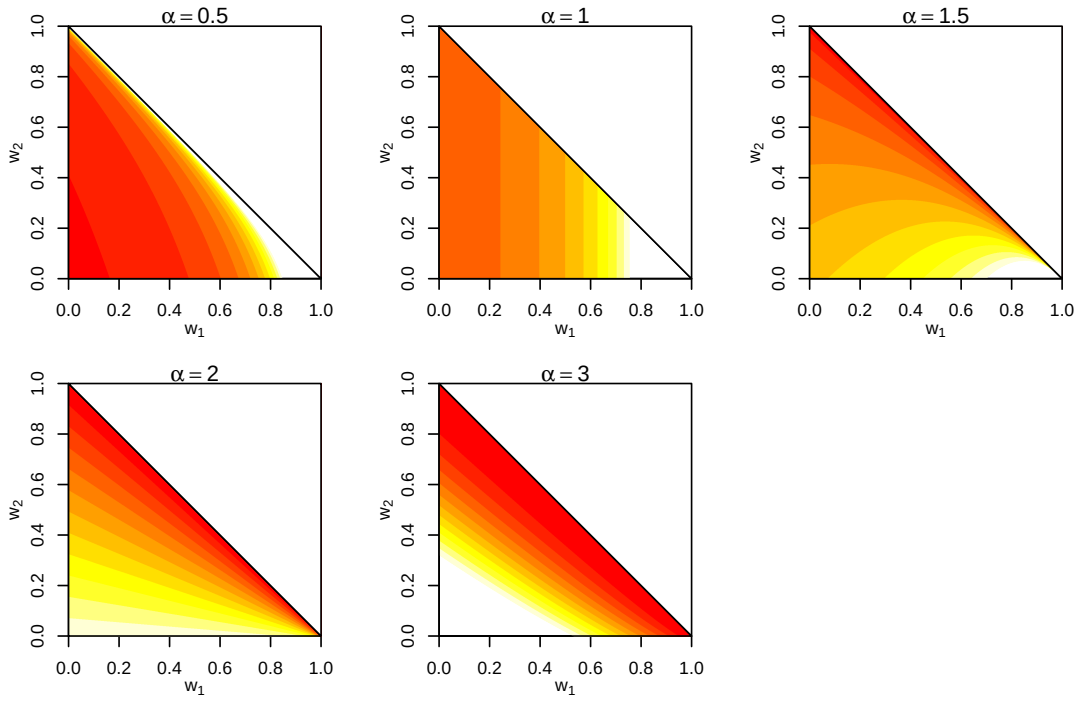

Figure 4.6: Sample-size-independent approach, size-biased. Conditional joint probability distribution $p(w_1, w_2 \mid \alpha)$, for different values of α. Red identifies small values while white identifies large.

Unconditionally, we have that:

$$p(w_1, w_2) = \int_0^\infty \alpha^2 \, \frac{(1 - w_1 - w_2)^{\alpha - 1}}{1 - w_1} \, \mathrm{d}p(\alpha \mid \boldsymbol{\eta}),$$

which, when $\alpha \sim \text{Ga}(a, b)$, leads to the following analytic expressions (see 4.3.2):

$$p(w_1, w_2) = a(a+1)\, b^a \, \frac{(b - \log(1 - w_1 - w_2))^{-a-2}}{(1 - w_1)(1 - w_1 - w_2)},$$
$$p(w_1) = ab^a \, \frac{(b - \log(1 - w_1))^{-a-1}}{(1 - w_1)},$$

which can potentially be used for further analytical considerations, when setting (a, b).

The plots in Figure 4.7 confirm that the joint unconditional distribution reflects a mix of the characteristics of the conditional joint distributions, in that the probability is amassed at the vertices $(0, 1)$ and $(1, 1)$, and along the edge that connects $(1, 0)$ and $(0, 0)$.

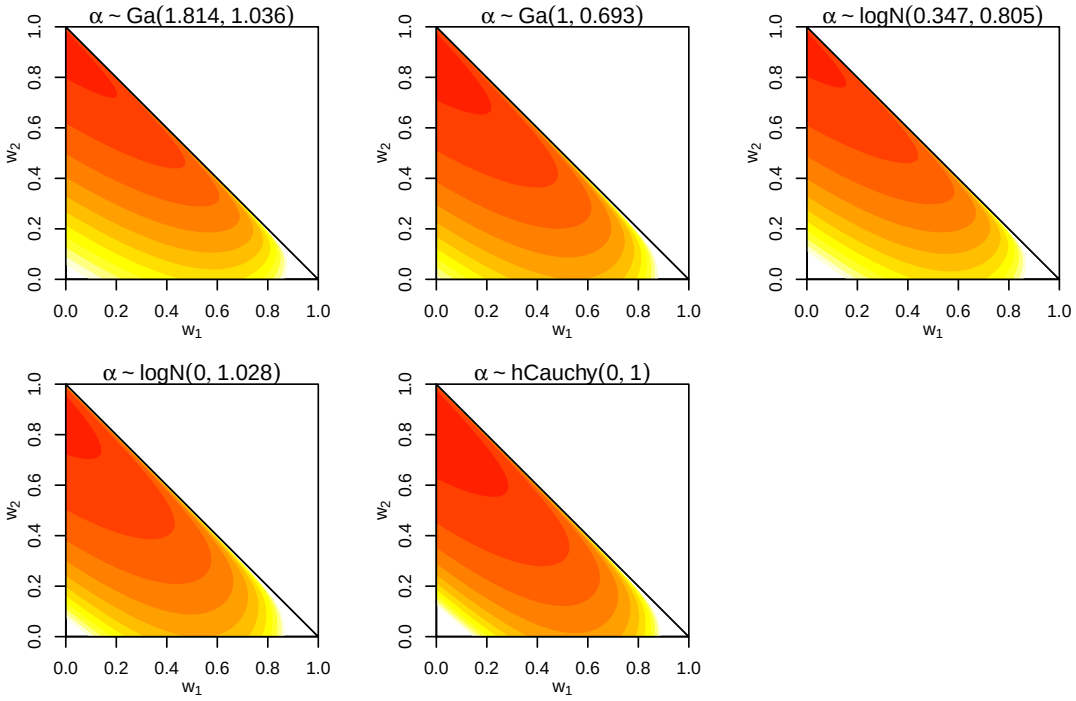

Figure 4.7: Sample-size-independent approach, size-biased. Joint probability distribution $p(w_1, w_2)$, for the distributional choices of α identified in table 4.4. Red identifies small values while white identifies large.

Distribution	$p(0<\alpha<1)$	$p(1<\alpha<2)$	$p(\alpha>2)$	η_1	η_2
Gamma	1/3	1/3	1/3	1.814	1.036
Gamma	1/2	1/4	1/4	1.000	0.693
Lognormal	1/3	1/3	1/3	0.347	0.805
Lognormal	1/2	1/4	1/4	0.000	1.028
Half-Cauchy	1/2	$p(\alpha>1)=1/2$		0.000	1.000

Table 4.4: Sample-size-independent approach, size-biased. A selection of results for different distributional choices of α, and for different choices of $p(0<\alpha<1)$, $p(1<\alpha<2)$, $p(\alpha>2)$.

Ranked weights

The same approach from section 4.3.2 can also be applied to ranked stick weights, in the Poisson-Dirichlet distribution representation (see section 2.1.3). We re-write equation 2.1.9 as follows:

$$p\left(w_1^{\downarrow},\ldots,w_r^{\downarrow}\mid\alpha\right)=\alpha^r\frac{\left(1-w_1^{\downarrow}-\ldots-w_r^{\downarrow}\right)^{\alpha-1}}{w_1^{\downarrow}\cdots w_r^{\downarrow}}\;F_\alpha\left(\frac{w_r^{\downarrow}}{1-w_1^{\downarrow}-\ldots-w_r^{\downarrow}}\right),\quad(4.3.5)$$

with $F_\alpha(x):=p\left(w_1^{\downarrow}\leq x\mid\alpha\right)$, which can be obtained by simulation. In the bivariate case, equation 4.3.5 becomes:

$$p\left(w_1^{\downarrow},w_2^{\downarrow}\mid\alpha\right)=\alpha^2\frac{\left(1-w_1^{\downarrow}-w_2^{\downarrow}\right)^{\alpha-1}}{w_1^{\downarrow}w_2^{\downarrow}}\;F_\alpha\left(\frac{w_2^{\downarrow}}{1-w_1^{\downarrow}-w_2^{\downarrow}}\right),\quad(4.3.6)$$

which is defined on

$$E=\left\{\left(w_1^{\downarrow},w_2^{\downarrow}\right):\left(w_1^{\downarrow}+w_2^{\downarrow}<1\right)\;\wedge\;\left(w_2^{\downarrow}<w_1^{\downarrow}\right)\right\}.$$

Since F_α is a cumulative probability distribution function, we have that

$$F_\alpha\left(\frac{w_2^{\downarrow}}{1-w_1^{\downarrow}-w_2^{\downarrow}}\right)=1,\quad\text{for } w_2^{\downarrow}\geq\frac{1}{2}-\frac{w_1^{\downarrow}}{2}.$$

As such, the restriction of equation 4.3.6 to

$$A=\left\{\left(w_1^{\downarrow},w_2^{\downarrow}\right):\left(w_1^{\downarrow}+w_2^{\downarrow}<1\right)\;\wedge\;\left(w_2^{\downarrow}<w_1^{\downarrow}\right)\;\wedge\;\left(w_2^{\downarrow}\geq\frac{1}{2}-\frac{1}{2}w_1^{\downarrow}\right)\right\}$$

yields

$$p\mid_A\left(w_1^{\downarrow},w_2^{\downarrow}\mid\alpha\right)=\alpha^2\frac{\left(1-w_1^{\downarrow}-w_2^{\downarrow}\right)^{\alpha-1}}{w_1^{\downarrow}w_2^{\downarrow}},\quad(4.3.7)$$

α	$p\mid_A \left(w_1^{\downarrow} \mid w_2^{\downarrow}, \alpha\right)$	$p\mid_A \left(w_2^{\downarrow} \mid w_1^{\downarrow}, \alpha\right)$
$0 < \alpha < 1$	convex; min. at $w_1^{\downarrow} = \frac{1-w_2^{\downarrow}}{2-\alpha}$	convex; min. at $w_2^{\downarrow} = \frac{1-w_1^{\downarrow}}{2-\alpha}$
$\alpha \geq 1$	decreasing	decreasing

Table 4.5: Sample-size-independent approach, ranked. Behaviour of $w_1^{\downarrow} \mid w_2^{\downarrow}, \alpha$ and $w_2^{\downarrow} \mid w_1^{\downarrow}, \alpha$, for different values of α, over A.

whose partial derivatives are easier to study than those of equation 4.3.6, and which bring some insights. In particular:

$$\frac{\partial p\mid_A \left(w_1^{\downarrow}, w_2^{\downarrow} \mid \alpha\right)}{\partial w_1^{\downarrow}} = \alpha^2 \frac{\left(1 - w_1^{\downarrow} - w_2^{\downarrow}\right)^{\alpha-2}}{w_1^{\downarrow 2} w_2^{\downarrow}} \left((2-\alpha)\, w_1^{\downarrow} + w_2^{\downarrow} - 1\right),$$
$$\frac{\partial p\mid_A \left(w_1^{\downarrow}, w_2^{\downarrow} \mid \alpha\right)}{\partial w_2^{\downarrow}} = \alpha^2 \frac{\left(1 - w_1^{\downarrow} - w_2^{\downarrow}\right)^{\alpha-2}}{w_1^{\downarrow} w_2^{\downarrow 2}} \left((2-\alpha)\, w_2^{\downarrow} + w_1^{\downarrow} - 1\right),$$

which leads to the considerations in table 4.5. Following the same approach we used in section 4.3.2, we also plot in Figure 4.8 the joint bivariate distribution of $p\left(w_1^{\downarrow}, w_2^{\downarrow} \mid \alpha\right)$ for key values of α, to spot any further visible patterns in the behaviour of $p\left(w_1^{\downarrow}, w_2^{\downarrow} \mid \alpha\right)$ as α varies.

The process for choosing the probability distribution $p(\alpha \mid \boldsymbol{\eta})$ and its parameters is analogous to the one in section 4.3.2. In Figure 4.9, we plot $p(w_1^{\downarrow}, w_2^{\downarrow})$ for different distributional choices of $p(\alpha \mid \boldsymbol{\eta})$, to show the impact of marginalising α out.

Unconditional size-biased weight distribution

As a side note, when $\alpha \sim \mathrm{Ga}(a, b)$, the joint distribution of the first H size-biased weights of a stick-breaking process has a simple closed analytical formula, once α is integrated out of equation 4.3.3:

$$\begin{aligned} p(w_1, \ldots, w_H) &= \int_0^{\infty} \frac{\alpha^H \left(1 - w_1 - \ldots - w_H\right)^{\alpha-1}}{(1-w_1)\ldots(1-w_1-\ldots-w_{H-1})} dp(\alpha) \\ &= \frac{\Gamma(\alpha+H)}{\Gamma(\alpha)}\, b^a\, \frac{\left[b - \log\left(1 - w_1 - \ldots - w_H\right)\right]^{-a-H}}{(1-w_1)\ldots(1-w_1-\ldots-w_H)} \qquad (4.3.8) \\ &\propto \frac{\left[b - \log\left(1 - w_1 - \ldots - w_H\right)\right]^{-a-H}}{(1-w_1)\,(1-w_1-\ldots-w_H)}. \qquad (4.3.9) \end{aligned}$$

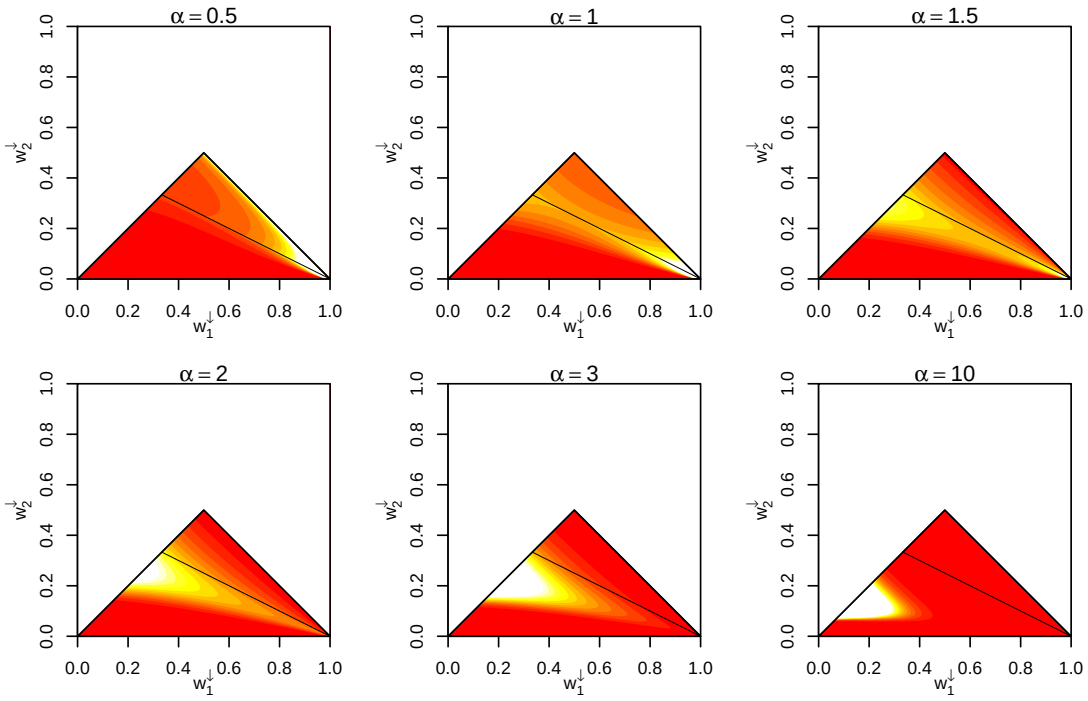

Figure 4.8: Sample-size-independent approach, ranked. Joint probability distribution $p\left(w_1^{\downarrow}, w_2^{\downarrow} \mid \alpha\right)$, for different values of α. The top right triangle identifies $A \subset E$. Red identifies small values while white identifies large.

The cumulative probability distribution of w_1 also has a simple explicit analytical representation, which played a role in the production of Figure 4.11:

$$p\left(w_1 \leq x\right) = \int_0^x p\left(w_1\right) \mathrm{d}w_1 = 1 - \left(\frac{b}{b - \log\left(1 - x\right)}\right)^a .$$

4.4 Discussion

In section 4.2 we studied the behaviour of SSD in view of the distribution that they induce on K_n, while in 4.3 we assessed SSI with respect to their implied stick-breaking distribution. Here we do the opposite, to cross-check how SSD and SSI behave in relation to each other's driving metric. For the sake of brevity, we only discuss size-biased stick breaking weights; conclusions with respect to ranked weights are similar.

We observe that the behaviour of the K_n-diffuse, the DORO and the quasi-degenerate[4]

[4]We parameterised the quasi-degenerate prior as a $\mathrm{Ga}\left(0.403, 0.370\right)$, to mirror the result from

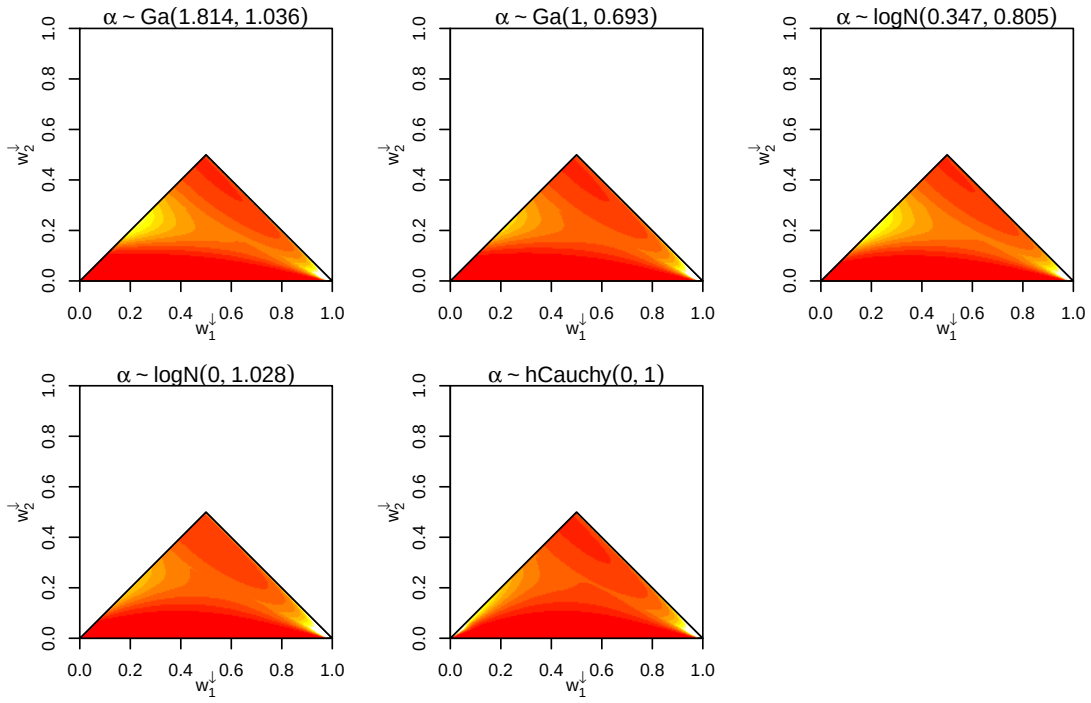

Figure 4.9: Sample-size-independent approach, ranked. Joint probability distribution $p\left(w_1^{\downarrow}, w_2^{\downarrow}\right)$, for the distributional choices of α identified in Table 4.4. Red identifies small values while white identifies large.

priors with respect to $p(w_1, w_2)$ (Figure 4.10, 4.11) is markedly different from the behaviour of SSI (Figure 4.7), and more extreme, as probability in SSD is by large concentrated at $(0, 0)$ or $(1, 0)$, while in SSI it is more spread out. SCAL is closer to SSI in this respect. We conclude that the K_n-diffuse and DORO priors would likely attract posterior estimates of (w_1, w_2) towards $(0, 0)$, while the quasi-degenerate prior would attract them towards $(1, 0)$. We also note that neither SCAL or the quasi-degenerate priors are diffuse in K_n (Figure 4.12).

Conversely, with respect to SSI, we consider the five test cases identified in Table 4.4, and we plot their implied distribution of K_n in Figure 4.13. We observe that their implied $p(K_n)$ is concentrated over values that are small, relative to n. We highlight that these five cases are just examples, as one's prior information to be reflected with SSI may well be entirely different from the cases in table 4.4.

the approximation method of SCAL, although we could have used any smaller value of (a, b). Conclusions would not materially change.

We more generally conclude that choices of $p(\alpha \mid \boldsymbol{\eta})$ that are diffuse in K_n are not necessarily diffuse in (w_1, w_2), and vice-versa. As such, users should assess whether SSD or SSI is more suitable with respect to their particular problem at hand; using both $p(K_n)$ and $p(w_1, w_2)$ as a reference to set $p(\alpha \mid \boldsymbol{\eta})$ could also be an option, if $p(K_n)$ is relevant to the application. We argue that SSI priors still hold a number of advantages over SSD, as outlined so far in the course of this thesis, and as we summarise in Section 4.6.

4.5 Case study: multiple DPMs

In this section we point to a setting where, by construction, the sample-size-dependent approach to choosing $p(\alpha \mid \boldsymbol{\eta})$ is inapplicable, while the sample-size-independent can still be relied upon. This was inspired by Müller and Rodriguez, 2013, figure 5.1.

Consider a partially exchangeable setting of J groups with n_j observations each, where $j = 1, \ldots, J$ and the groups share the same common underlying precision parameter α. This framework, while extremely simple, allows information to be borrowed between groups through their dependence on a common α.

We denote the observations by $y_{i,j},\ i = 1, \ldots, n_j,\ j = 1, \ldots, J$, and the random number of clusters in group j by $K_{n,j}$. The model is:

$$
\begin{aligned}
y_{i,j} \mid \theta_{i,j} &\sim p(y_{i,j} \mid \theta_{i,j}) \\
\theta_{i,j} \mid G_j &\sim G_j \\
G_j \mid \alpha &\sim \mathrm{DP}(\alpha, G_{0j}) \\
\alpha &\sim p(\alpha \mid \boldsymbol{\eta}).
\end{aligned}
$$

It is clear that, by construction, multiple $K_{n,j}$ are involved, hence there is no unique probability distribution of K_n and no unique sample size n that can be used as a target for the sample-size-independent approaches that we have described. Conversely, our sample-size-independent approach is still applicable.

4.6 Conclusions

In section 4.2 we highlighted the limitations of previous sample-size-dependent approaches, which leverage the implied distribution of K_n in order to make decisions about $p(\alpha \mid \boldsymbol{\eta})$. Their limitations include:

- dependence on n. All SSD priors are only optimal for one particular value of n, and they need to be updated as n varies;
- unmet assumptions. DORO minimises the Kullback-Leibler distance between $p(K_n)$ and the discrete uniform distribution, however the discrete uniform target can never be attained, as $p(K_n)$ converges to a negative binomial for $n \to \infty$ (see section 4.2.1). The assumption that a non-informative prior would reflect as a discrete uniform distribution induced on $p(K_n)$ is not reflected by Jeffreys' prior either, which has a very different shape from discrete uniform;
- asymptotic breakdown. The approximation suggested by SCAL, in relation to their test case, as an alternative to their fully-fledged approach is only valid over a narrow range of values of n; for moderately large n, it results in $\alpha \sim \mathrm{Ga}(a, b)$, $(a, b) \approx (0, 0)$. This is undesirable, as it implies $p(K_n = 1) \approx 1$, which negates the reason to use a nonparametric model in the first place (see sections 4.2.1, 4.2.2);
- over-informativeness. Although K_n-diffuse and DORO priors are diffuse in K_n, they are very informative with respect to (w_1, w_2) and $\left(w_1^{\downarrow}, w_2^{\downarrow}\right)$ (see section 4.4);
- inapplicability. SSD priors are inapplicable when multiple DPMs with different sample sizes share the same α (see section 4.5).

In section 4.3 we introduced a new approach, which is based on the appraisal of the implied joint distribution of the stick-breaking weights instead (either in size-biased order, or ranked). This is equivalent to thinking in terms of the asymptotic relative

cluster sizes, for $n \to \infty$. Our approach does not suffer from the aforementioned limitations, although we acknowledge that SSD priors still have a place, depending on the specific nature of the problem at hand, and that potentially both $p(K_n)$ and the stick breaking weights could be concurrently leveraged to inform one's choice of $p(\alpha \mid \boldsymbol{\eta})$.

Out of the two alternatives that we propose, the one that is based on $p(w_1, w_2)$ appears to be easiest to interpret, compute and exercise one's judgement on, due to the distinctive behaviour of $p(w_1, w_2 \mid \alpha)$ for various levels of α (Figure 4.6) and to the availability of easier analytical formulae. Quantitative measurements can be carried out, as exemplified in Table 4.4, to determine how to mix precisely over those behaviours. A simple exact formula is also available, when $\alpha \sim \text{Exp}(\eta)$ (see equation 4.3.4). We envisage use of SSI priors by practitioners as a valid alternative to SSD priors, and possibly more useful in certain situations.

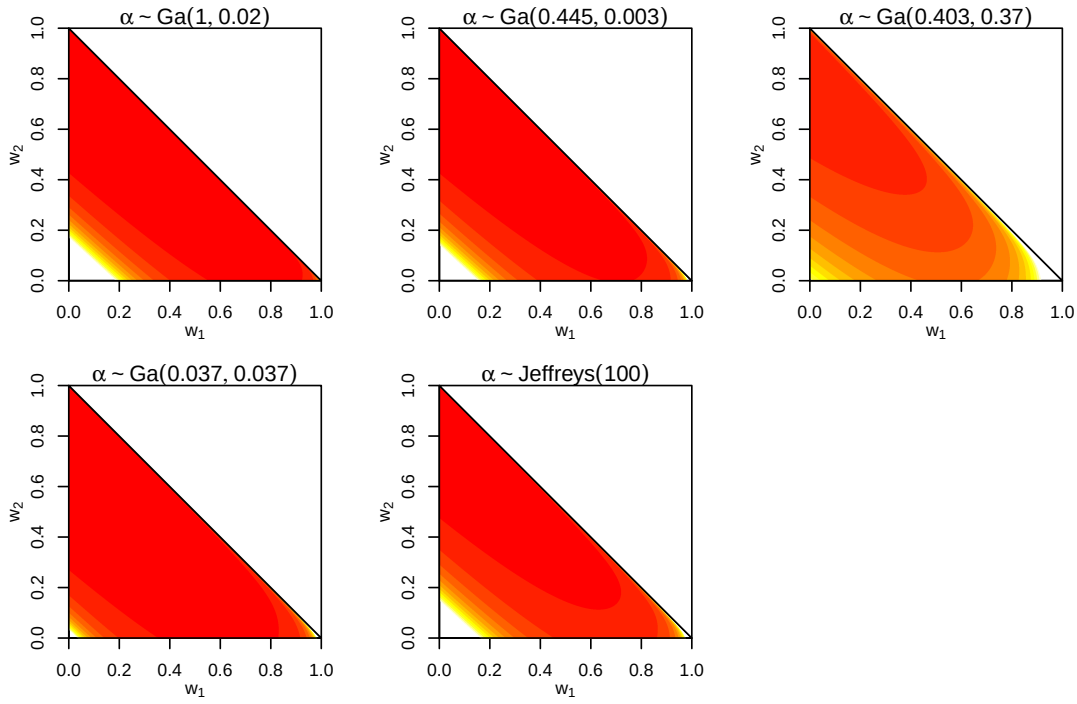

Figure 4.10: Size-biased joint probability distribution $p(w_1, w_2)$ underlying the K_n-diffuse, DORO, SCAL, quasi-degenerate and Jeffreys' priors for $n = 100$. Red identifies small values while white identifies large.

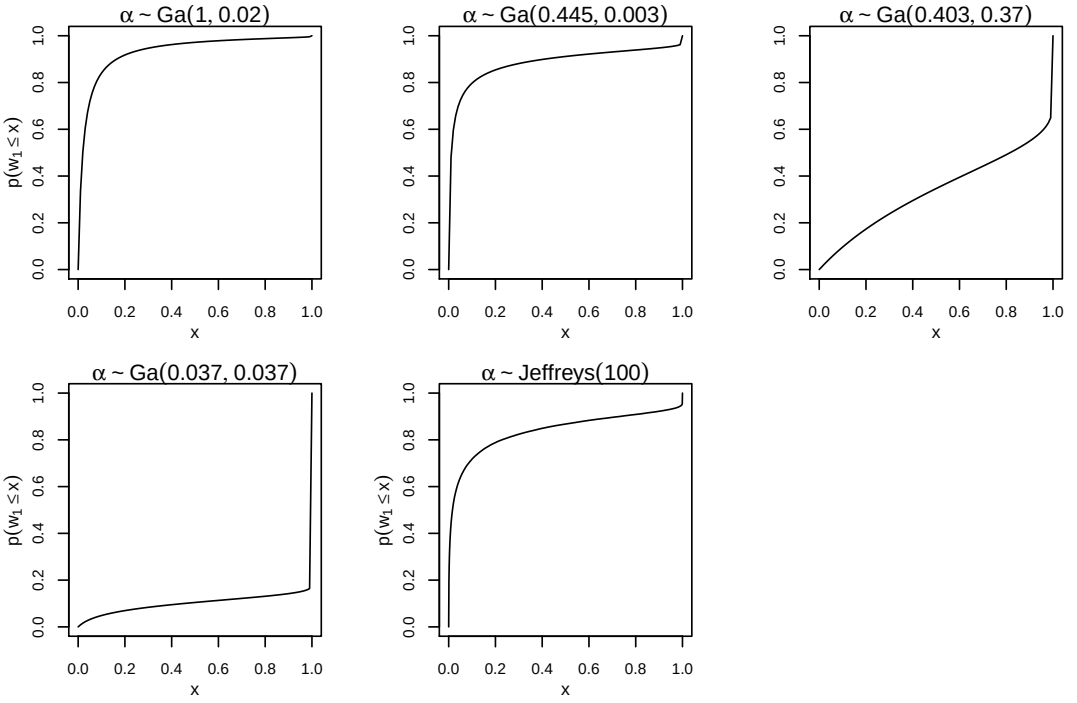

Figure 4.11: Size-biased cumulative probability distribution of w_1 underlying the K_n-diffuse, DORO, SCAL, quasi-degenerate and Jeffreys' priors for $n = 100$.

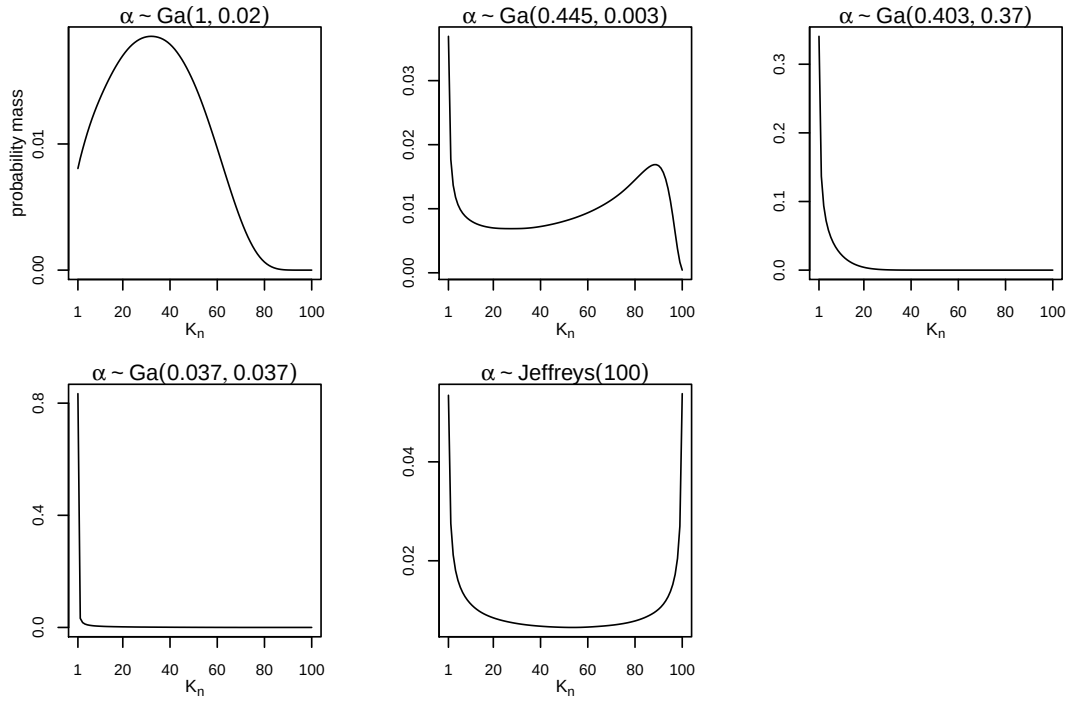

Figure 4.12: Prior distribution $p(K_n)$ induced by $p(\alpha)$, for the K_n-diffuse, DORO, SCAL, quasi-degenerate and Jeffreys' priors for $n = 100$.

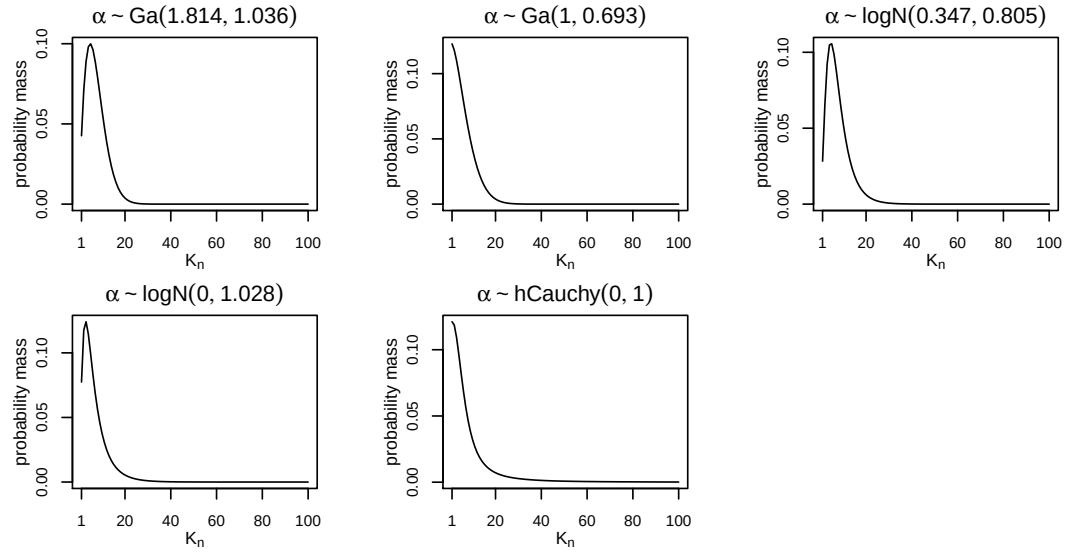

Figure 4.13: Prior distribution $p(K_n)$ induced by $p(\alpha)$, for $n = 100$, for the distributional choices of α identified in Table 4.4.

Chapter 5

Labels-switching moves via Metropolis jumps

As outlined in Chapter 3, inferential algorithms for Dirichlet Process Mixtures based on the Gibbs sampler can be distinguished into *collapsed* samplers and *stick-breaking* samplers. The former integrate G out, and therefore remove the need to draw from the infinite-dimensional random probability measure G, hence the term *collapsed*; the latter instead move from the stick-breaking representation of G, and model it explicitly. These two types of samplers are also known as *marginal* and *conditional*, respectively.

While stick-breaking samplers allow to make inferences about parameters which the collapsed samplers skip altogether, such as $\boldsymbol{r}$ and $\boldsymbol{w}$, for the same reason they are also slower at mixing, as they have a wider parameter space to explore. Their parameter space is also often multi-modal (Papaspiliopoulos & Roberts, 2008), due to the fact that the algorithm attempts to identify which of the sticks in the stick-breaking construction process each observation precisely originates from, and the probabilities of competing configurations are often similar. To mitigate their slow mixing, stick-breaking inferential algorithms have been enhanced with steps which attempt to jump between modes via Metropolis-Hastings moves where various parameters, including the cluster labels, are switched.

In this chapter, to address the slow mixing issue, we first revisit three label-switching Metropolis moves from literature: *move 1* and *move 2* from Papaspiliopoulos and Roberts, 2008, and *move 3* from Hastie et al., 2015. We re-derive them, we show how the published formula of move 3 was incomplete, and we complete it, so that it can lead to correct results. We adapt move 1 and 2 from the retrospective sampler framework to the slice sampler framework. We also propose an improvement, consisting of two Metropolis-Hastings acceptance ratio adjustments and a new selection approach to their mixture of kernels, to both move 2 and move 3, so that they can be attempted more frequently, even in Gibbs scans where the conditional number of clusters is 1. We then introduce move 4, which in our view constitutes a more intuitive alternative to move 3 in the parameters that it switches, and which performs similarly to our improved version of move 3.

5.1 Hybrid MCMC algorithms and deterministic moves

All of the label-switching moves described in this chapter are Metropolis-within-Gibbs mixtures of deterministic transition kernels.

They are *mixtures* of transition kernels because, as their name implies, they randomly select which transition kernel to make a proposal from, amongst a wider set of pre-defined kernels. Each individual kernel targets two pre-specified clusters, and either switches some of their parameters (moves 1, 2 and 4), or re-calculates them (move 3). While the moves do not explicitly assign a probability mass to each component of the mixture, to randomly draw one, the way they operate is entirely equivalent to doing so, as the mapping between the set of all possible pairs of randomly selected clusters and the mixture components is a bijection. The cluster pairs are selected uniformly at random from a pre-defined finite set, which for move 1 only involves occupied clusters, while for moves 2, 3 and 4 involves both occupied and unoccupied

clusters.

They are *deterministic* because, once a kernel is randomly selected from the mixture, the formulation of the proposal does not entail randomness: rather, the proposal involves either swapping some pre-determined parameter values, or re-calculating them according to a pre-defined rule. While unusual, this is not unheard of, and in fact the reversible jump MCMC algorithm also uses deterministic Metropolis jumps (Green, 1995). The use of deterministic moves requires the use of the Jacobian determinant (Hastie & Green, 2012, page 314) to calculate their Metropolis acceptance ratio.

They are *Metropolis-within-Gibbs* because they involve interleaving a fully-functional Gibbs sampler with a Metropolis-Hastings move. Because these invariant Metropolis moves are interleaved with a wider, irreducible Gibbs sampler, the overall chain produced by the composite algorithm remains irreducible even though none of the individual Metropolis kernels or mixtures thereof is (Tierney, 1994, section 2.4).

5.2 Move 1

Move 1 and move 2 were introduced in Papaspiliopoulos and Roberts, 2008 in the context of the retrospective sampler, and here we provide their derivation (which is not included in Papaspiliopoulos and Roberts, 2008), so that we can adapt them to the slice-sampler framework of Papaspiliopoulos, 2008. These moves were also used in Yau and Holmes, 2011 and Hastie et al., 2015. Move 1 is a mixture of transition kernels; it randomly selects two clusters, and it switches their labels in $\boldsymbol{r}$ and their locations in $\boldsymbol{m}$. As reflected by its acceptance ratio, it may switch distant clusters, and it is more easily accepted for clusters with similar size and weight. Move 1 consists of the following steps, at the end of each Gibbs iteration:

1. if $k > 1$, draw s and t uniformly at random from the k unique values in $(r_1 \ldots, r_n)$, without replacement, otherwise skip this and all following steps and move to the next Gibbs iteration;

2. switch the corresponding two elements of $\boldsymbol{m}$:

$$m'_s = m_t$$
$$m'_t = m_s$$

3. switch the values in $\boldsymbol{r}, \forall i \in \{1, \ldots, n\}$:

$$r'_i = \begin{cases} t, \text{ if } i \in \{i : r_i = s\} \\ s, \text{ if } i \in \{i : r_i = t\} \\ r_i, \text{ otherwise} \end{cases}$$

This move is symmetrical, as

$$m''_s = m_s$$
$$m''_t = m_t$$
$$r''_i = r_i, \quad i = 1, \ldots, n$$

for every transition kernel in the mixture.

The full conditional of the parameters being updated is $\boldsymbol{r}, \boldsymbol{m}$ is:

$$\begin{aligned} p(\boldsymbol{r}, \boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{w}) &\propto p(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{m}, \boldsymbol{w}) \cdot p(\boldsymbol{r} \mid \boldsymbol{w}, \boldsymbol{m}) \cdot p(\boldsymbol{m} \mid \boldsymbol{w}) \\ &\propto p(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{m}) \cdot p(\boldsymbol{r} \mid \boldsymbol{w}) \cdot p(\boldsymbol{m}) . \end{aligned}$$

When calculating its Metropolis acceptance ratio, the first and third factors above cancel out, leading to:

$$\begin{aligned} R &= \min\left\{1, \frac{p(\boldsymbol{r}', \boldsymbol{m}' \mid \boldsymbol{y}, \boldsymbol{w})}{p(\boldsymbol{r}, \boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{w})}\right\} = \min\left\{1, \frac{p(\boldsymbol{r}' \mid \boldsymbol{w})}{p(\boldsymbol{r} \mid \boldsymbol{w})}\right\} \\ &= \min\left\{1, \frac{w_s^{n_t} w_t^{n_s}}{w_t^{n_t} w_s^{n_s}}\right\} = \min\left\{1, \left(\frac{w_s}{w_t}\right)^{n_t - n_s}\right\} . \end{aligned}$$

We note that, because all of the moves entailed are pure parameters swaps (rather than transformations thereof), the absolute value of their Jacobian determinant is 1, hence it is omitted from the derivation of the acceptance ratio above, for brevity.

5.2.1 Slice sampler adaptation

Move 1 as just described was initially proposed in the context of the retrospective sampler (Papaspiliopoulos & Roberts, 2008), but the retrospective sampler later evolved and was combined with the work of (Walker, 2007), to constitute what we refer to in this thesis as the *slice sampler* (Papaspiliopoulos, 2008).

As the slice sampler uses one additional latent variable ($\boldsymbol{u}$) which is absent from the retrospective sampler, move 1 requires some adaptation in order to be used with it. To preserve the convergence of the chain to the right stationary distribution, one possible solution is to condition move 1 to $\boldsymbol{u}$, thereby sampling from $p(\boldsymbol{r}, \boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{w}, \boldsymbol{u})$; alternatively, another solution would be to extend the move by adding $\boldsymbol{u}$ to the parameters subject to the Metropolis-Hastings jump, therefore sampling from $p(\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{u} \mid \boldsymbol{y}, \boldsymbol{w})$.

The approach that conditions on $\boldsymbol{u}$ does not work well. The full conditional becomes:

$$\begin{aligned} p(\boldsymbol{r}, \boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{w}, \boldsymbol{u}) &\propto p(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{m}, \boldsymbol{w}, \boldsymbol{u}) \cdot p(\boldsymbol{r} \mid \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{u}) \cdot p(\boldsymbol{m} \mid \boldsymbol{w}, \boldsymbol{u}) \\ &\propto p(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{m}) \cdot p(\boldsymbol{r} \mid \boldsymbol{w}, \boldsymbol{u}) \cdot p(\boldsymbol{m}), \end{aligned} \tag{5.2.1}$$

and, as previously, the first and third factors cancel out in the acceptance ratio, leading to:

$$\begin{aligned} R &= \min\left\{1, \frac{p(\boldsymbol{r}', \boldsymbol{m}' \mid \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{w})}{p(\boldsymbol{r}, \boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{w})}\right\} = \min\left\{1, \frac{p(\boldsymbol{r}' \mid \boldsymbol{w}, \boldsymbol{u})}{p(\boldsymbol{r} \mid \boldsymbol{w}, \boldsymbol{u})}\right\} \\ &= \min\left\{1, \left\{\prod_{\{i:r_i'=t\}} \mathbb{I}_{0,w_t}(u_i)\right\} \cdot \left\{\prod_{\{j:r_j'=s\}} \mathbb{I}_{0,w_s}(u_j)\right\}\right\} \end{aligned} \tag{5.2.2}$$

because

$$p(r_i = h \mid \boldsymbol{w}, u_i) \propto p(u_i \mid \boldsymbol{w}, r_i) \cdot p(r_i \mid \boldsymbol{w}) = \frac{1}{w_h} \cdot \mathbb{I}_{0,w_h}(u_i) \cdot w_h,$$

hence

$$p(r_i = h \mid \boldsymbol{w}, u_i) = \frac{p(u_i, r_i = h \mid \boldsymbol{w})}{p(u_i \mid \boldsymbol{w})} = \frac{\mathbb{I}_{0,w_h}(u_i)}{\sum_h \mathbb{I}_{0,w_h}(u_i)}.$$

We observe that the acceptance ratio is a product of $n_s + n_t$ indicator functions, and

that, as n increases, acceptance becomes less likely, as it becomes more likely that at least one indicator function returns 0. We therefore do not use this approach in the remainder of this thesis.

The approach which updates $\boldsymbol{u}$ concurrently to $\boldsymbol{r}, \boldsymbol{m}$ works better, and it is also easy to apply. It suffices to observe that

$$p\left(\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{u} \mid \boldsymbol{y}, \boldsymbol{w}\right) = p\left(\boldsymbol{r}, \boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{w}\right) \cdot p\left(\boldsymbol{u} \mid \boldsymbol{r}, \boldsymbol{m}, \boldsymbol{y}, \boldsymbol{w}\right), \tag{5.2.3}$$

and that the first factor corresponds to the Metropolis move ignoring $\boldsymbol{u}$, while the second factor is exactly the probability measure used to resample $\boldsymbol{u} \mid \ldots$ in the slice sampler algorithm. Therefore, in the slice sampler framework, it is legitimate to apply Metropolis-Hastings move 1 exactly as in the retrospective sampler framework (i.e. using the same acceptance ratio), provided that one also subsequently resamples $\boldsymbol{u} \mid \boldsymbol{r}, \boldsymbol{m}, \boldsymbol{y}, \boldsymbol{w}$ as per equation 3.4.9. The same considerations apply to the moves that we describe in the following sections, and for brevity we do not repeat them in the remainder.

5.3 Move 2

While move 1 attempts to switch the labels ($\boldsymbol{r}$) and the locations ($\boldsymbol{m}$) of two randomly selected occupied clusters, it does not switch the length of their respective originating sticks ($\boldsymbol{w}$). In our tests (see section 5.7), and as also reflected in Hastie et al., 2015, this leads to a relatively low acceptance ratio, and only to a limited improvement in the mixing speed of the chain. Move 2 instead attempts to switch the labels of two juxtaposed clusters ($\boldsymbol{r}$), their locations ($\boldsymbol{m}$) and, while it does not switch their corresponding weights $\boldsymbol{w}$, it does switch two contiguous elements of $\boldsymbol{v}$, which in turn are the building blocks of $\boldsymbol{w}$.

Move 2 is a mixture of transition kernels, and it proceeds as follows:

- draw s uniformly at random[1], to select clusters s and $s+1$;

- switch the corresponding elements of $\boldsymbol{m}$:

$$m'_s = m_{s+1}$$
$$m'_{s+1} = m_s$$

- switch the values in $\boldsymbol{r}, \forall i \in \{1, \ldots, n\}$:

$$r'_i = \begin{cases} s+1, \text{ for } i \in \{i : r_i = s\} \\ s, \text{ for } i \in \{i : r_i = s+1\} \\ r_i, \text{ otherwise} \end{cases}$$

- switch (v_s, v_{s+1}) (and re-calculate $\boldsymbol{w}$):

$$v'_s = v_{s+1}$$
$$v'_{s+1} = v_s$$

If s is drawn uniformly at random from a fixed, finite set $\{1, \ldots, T\}$, then this move is symmetrical (however please see our considerations in Section 5.3.2), as

$$m''_s = m_s$$
$$m''_{s+1} = m_{s+1}$$
$$r''_i = r_i, \quad i = 1, \ldots, n$$
$$v''_s = v_s$$
$$v''_{s+1} = v_{s+1}$$

for every transition kernel in the mixture.

The full conditional of the parameter vectors being updated is:

$$p(\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{v} \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{m}, \boldsymbol{v}) \cdot p(\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{v})$$

[1]See our considerations in Section 5.3.2.

$$
\begin{aligned}
&= p\left(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{m}\right) \cdot p\left(\boldsymbol{m}\right) \cdot p\left(\boldsymbol{r}, \boldsymbol{v}\right) \\
&= p\left(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{m}\right) \cdot p\left(\boldsymbol{m}\right) \cdot p\left(\boldsymbol{r} \mid \boldsymbol{v}\right) \cdot p\left(\boldsymbol{v}\right).
\end{aligned}
$$

The first two factors above cancel out when calculating the acceptance ratio, which ultimately is equal to:

$$
\begin{aligned}
R &= \min\left\{1, \gamma \cdot \frac{p\left(\boldsymbol{r}' \mid \boldsymbol{v}'\right)}{p\left(\boldsymbol{r} \mid \boldsymbol{v}\right)} \cdot \frac{p\left(\boldsymbol{v}'\right)}{p\left(\boldsymbol{v}\right)} = \frac{p\left(\boldsymbol{r}' \mid \boldsymbol{w}'\right)}{p\left(\boldsymbol{r} \mid \boldsymbol{w}\right)} \cdot \frac{p\left(v'_s\right) \cancel{p\left(v'_{s+1}\right)}}{\cancel{p\left(v_s\right)} \, p\left(v_{s+1}\right)}\right\} \\
&= \min\left\{1, \gamma \cdot \left(\prod_{i:r_i=s} \frac{\left(1 - v_{s+1}\right) v_s}{v_s}\right) \left(\prod_{j:r_j=s+1} \frac{v_{s+1}}{\left(1 - v_s\right) v_{s+1}}\right)\right\} \\
&= \min\left\{1, \gamma \cdot \frac{\left(1 - v_{s+1}\right)^{n_s}}{\left(1 - v_s\right)^{n_{s+1}}}\right\},
\end{aligned}
$$

where γ is an adjustment that we outline in Section 5.3.2. As in move 1, because all of the moves entailed are pure parameters swaps (rather than transformations thereof), the absolute value of their Jacobian determinant is 1, hence it is omitted from the derivation of the acceptance ratio above, for brevity.

5.3.1 Slice sampler adaptation

Just like in Section 5.2.1, here too we recommend to update $\boldsymbol{u}$ along with the other parameters refreshed by move 2, rather than to condition move 2 to $\boldsymbol{u}$. The full conditional is:

$$
p\left(\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{v}, \boldsymbol{u} \mid \boldsymbol{y}\right) = p\left(\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{v} \mid \boldsymbol{y}\right) \cdot p\left(\boldsymbol{u} \mid \boldsymbol{r}, \boldsymbol{m}, \boldsymbol{v}, \boldsymbol{y}\right),
$$

meaning that, in the slice sampler framework, move 2 can be applied exactly as in the retrospective sampler framework, provided that $\boldsymbol{u} \mid \boldsymbol{r}, \boldsymbol{m}, \boldsymbol{v}, \boldsymbol{y}$ is also resampled as per equation 3.4.9.

5.3.2 Cluster selection

The original paper where move 2 was introduced instructs to select s at random, but it does not indicate how to best select it, which is not entirely trivial as there is an

infinite number of clusters in the Dirichlet process, hence no practical way of doing so.

Inspection of three software implementations of move 2 suggests the following:

- the code accompanying the original article where move 2 was introduced does not seem to implement move 2, but only move 1 (Papaspiliopoulos & Roberts, 2008), hence it offers no clarification in this respect;
- the code accompanying Yau and Holmes, 2011 appears to be equivalent to selecting s amongst $\{1, \ldots, \max(\boldsymbol{r}) - 1\}$;
- the code accompanying Hastie et al., 2015 also appears to be equivalent to selecting s amongst $\{1, \ldots, \max(\boldsymbol{r}) - 1\}$. This is also how the same authors describe their move 3 in Hastie et al., 2015, and move 3 is supposed to mirror move 2 in how it selects the two clusters to switch.

We see two issues with sampling s from $\{1, \ldots, \max(\boldsymbol{r}) - 1\}$:

- it does not lead to any proposal, and therefore cannot trigger any move, for Gibbs scans where the conditional number of clusters is $k = 1$, which may lead to a loss of efficiency, as it means the move is attempted less often than once every Gibbs iteration;
- it does not ensure meeting the detailed balance condition.

The reason why it does not meet the detailed balance condition is that $\max(\boldsymbol{r}) - 1$ may change as a result of the move being accepted, thereby altering the probability of coming back from $\boldsymbol{r}'$ to $\boldsymbol{r}$. For example, when $\boldsymbol{r} = (1, 2, 4)$, there is $1/3$ probability that the sub-kernel corresponding to $s = 3$ is selected from the mixture, which if accepted would lead to $\boldsymbol{r}' = (1, 2, 3)$. But then, to move back from $\boldsymbol{r}' = (1, 2, 3)$ to $\boldsymbol{r}'' = (1, 2, 4)$, the probability of the right kernel being selected from the mixture would be $1/2$, leading to the need for a Metropolis-Hastings adjustment to the acceptance ratio equal to $s/(s-1)$.

More generally, write the acceptance ratio of move 2 as:

$$R = \min\left\{1, \gamma \cdot \frac{(1-v_{s+1})^{n_s}}{(1-v_s)^{n_{s+1}}}\right\},$$

where by γ we indicate an adjustment factor. By using the appropriate adjustment factor, not only sampling s from $\{1, \ldots, \max(\boldsymbol{r}) - 1\}$ becomes feasible, but so does sampling from $\{1, \ldots, \max(\boldsymbol{r})\}$ too. We argue that sampling from the latter is more appealing, as it does return a proposal and may trigger a switch even for Gibbs scans with $k = 1$ (and, depending on the application, the proportion of cases where $k = 1$ may potentially be significant). When sampling s from $\{1, \ldots, \max(\boldsymbol{r})\}$, the following γ adjustments are required:

$$\gamma = \begin{cases} \frac{s}{s+1}, & \text{if } s = \max(\boldsymbol{r}), \\ \frac{s+1}{s}, & \text{if } s = \max(\boldsymbol{r}) - 1 \text{ and } \#\{i : r_i = s\} = 0, \\ 1, & \text{otherwise.} \end{cases}$$

A possibly easier solution than the one above is to set a fixed threshold $T \in \mathbb{N}^+$, so that s is picked uniformly at random from $\{1, \ldots, T\}$ – this also guarantees detailed balance with $\gamma = 1$, without the need for the aforementioned adjustments. However, this solution requires some care in setting up the sampler, to ensure that the wider Gibbs algorithm always produces at least $T + 1$ elements of $\boldsymbol{w}, \boldsymbol{m}, \ldots$ at each scan. This alternate solution is not equivalent to selecting s from $\{1, \ldots, \max(\boldsymbol{r})\}$ though, in that by lacking adaptation to the length of $\boldsymbol{r}$, it may end up sampling many clusters which are empty, or in not sampling occupied clusters of order higher than T, which may miss out on more useful switch proposals.

In section 5.7 we test the move with and without our two adjustments to the acceptance ratio.

5.4 Move 3

The idea of move 3 is to "simultaneously propose an update of the new cluster weights so they are something like their expected value conditional upon their allocations" (Hastie et al., 2015).

Move 3 is a mixture of transition kernels, and it consists of the following steps:

- if $k > 1$, set $r_{\max} = \max_{1 \leq i \leq n} r_i$ and draw s uniformly at random from $\{1, \ldots, r_{\max} - 1\}$, to select clusters s and $s+1$, otherwise skip this and all following steps and move to the next Gibbs iteration (however, we suggest the modification that we outline in Section 5.3.2, so that the condition $k > 1$ can be dropped, and so that s can be sampled from $\{1, \ldots, r_{\max}\}$ instead);
- switch the labels of clusters s and $s+1$ as follows, for $i = 1, \ldots, n$:

$$r_i' = \begin{cases} s+1, \text{ for } i \in \{i : r_i = s\} \\ s, \text{ for } i \in \{i : r_i = s+1\} \\ r_i, \text{ otherwise} \end{cases}$$

- switch their locations as follows:

$$m_s' = m_{s+1}$$
$$m_{s+1}' = m_s$$

- update v_s, v_{s+1} as follows:

$$v_s' = \frac{w_s'}{\prod_{l<s}(1 - v_l)} \tag{5.4.1}$$
$$v_{s+1}' = \frac{w_{s+1}'}{(1 - v_s')\prod_{l<s}(1 - v_l)} \tag{5.4.2}$$

  where in turn

$$w_s' = w_{s+1}\frac{w_s + w_{s+1}}{W'}R_1 \tag{5.4.3}$$
$$w_{s+1}' = w_s\frac{w_s + w_{s+1}}{W'}R_2 \tag{5.4.4}$$

and the following abbreviations are in place:

$$R_1 = \frac{\mathbb{E}\left[w_s \mid \boldsymbol{r}'\right]}{\mathbb{E}\left[w_{s+1} \mid \boldsymbol{r}\right]} = \frac{1+\alpha+n_{s+1}+\sum_{l>s+1} n_l}{\alpha+n_{s+1}+\sum_{l>s+1} n_l} \tag{5.4.5}$$

$$R_2 = \frac{\mathbb{E}\left[w_{s+1} \mid \boldsymbol{r}'\right]}{\mathbb{E}\left[w_s \mid \boldsymbol{r}\right]} = \frac{\alpha+n_s+\sum_{l>s+1} n_l}{1+\alpha+n_s+\sum_{l>s+1} n_l} \tag{5.4.6}$$

$$W' = w_{s+1}R_1 + w_s R_2. \tag{5.4.7}$$

Although each individual transition sub-kernel is symmetrical, the overall move 3 as described in Hastie et al., 2015 requires the adjustments we outline in Section 5.3.2, unless for example a modification is made so that s is selected from a fixed set $\{1, \ldots, T\}$.

The acceptance ratio reported in Hastie et al., 2015 is:

$$R = \min\left\{1, \left(\frac{w_s + w_{s+1}}{w_{s+1}R_1 + w_s R_2}\right)^{n_s+n_{s+1}} R_1^{n_{s+1}} R_2^{n_s}\right\}, \tag{5.4.8}$$

which however is incomplete and therefore equation 5.4.8 is inaccurate; we derive the complete acceptance ratio formula in the next subsection (see specifically equation 5.4.12). Contrary to move 1 and move 2, the move entailing v_s, v_{s+1} is not a pure parameter swap, but it involves a transformation where the absolute value of its Jacobian determinant is not guaranteed to be 1.

5.4.1 Acceptance ratio

To derive the acceptance ratio for move 3, we write the full conditional

$$p\left(\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{v} \mid \boldsymbol{y}\right) \propto p\left(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{m}\right) \cdot p\left(\boldsymbol{m}\right) \cdot p\left(\boldsymbol{r} \mid \boldsymbol{v}\right) \cdot p\left(\boldsymbol{v}\right). \tag{5.4.9}$$

In case that the additional variable $\boldsymbol{u}$ (absent from the equation above) is of interest, it can well be added in with exactly the same blocking approach as in section 5.3.1).

In equation 5.4.9, the first two factors cancel out, when calculating the acceptance ratio. What remains to be calculated is therefore:

$$\frac{p\left(\boldsymbol{r}' \mid \boldsymbol{v}'\right)}{p\left(\boldsymbol{r} \mid \boldsymbol{v}\right)} \cdot \frac{p\left(\boldsymbol{v}'\right)|J|}{p\left(\boldsymbol{v}\right)}. \tag{5.4.10}$$

The first ratio in equation 5.4.10 is equal to:

$$\begin{aligned}\frac{p\left(\boldsymbol{r}' \mid \boldsymbol{v}'\right)}{p\left(\boldsymbol{r} \mid \boldsymbol{v}\right)} &= \frac{{w'_s}^{n'_s}}{w_s^{n_s}} \frac{{w'_{s+1}}^{n'_{s+1}}}{{w_{s+1}}^{n_{s+1}}} = \frac{{w'_s}^{n_{s+1}}}{w_s^{n_s}} \frac{{w'_{s+1}}^{n_s}}{{w_{s+1}}^{n_{s+1}}} = \left(\frac{w'_{s+1}}{w_s}\right)^{n_s} \left(\frac{w'_s}{w_{s+1}}\right)^{n_{s+1}} \\ &= \left(\frac{\cancel{w_s} \frac{w_s+w_{s+1}}{W'} \frac{\mathbb{E}[w_{s+1}|\boldsymbol{r}']}{\mathbb{E}[w_s|\boldsymbol{r}]}}{\cancel{w_s}}\right)^{n_s} \cdot \left(\frac{\cancel{w_{s+1}} \frac{w_s+w_{s+1}}{W'} \frac{\mathbb{E}[w_s|\boldsymbol{r}']}{\mathbb{E}[w_{s+1}|\boldsymbol{r}]}}{\cancel{w_{s+1}}}\right)^{n_{s+1}} \\ &= \left(\frac{w_s + w_{s+1}}{w_{s+1}R_1 + w_s R_2}\right)^{n_s+n_{s+1}} R_2^{n_s} R_1^{n_{s+1}}.\end{aligned}$$

which matches the calculations from Hastie et al., 2015.

The calculation of the second ratio in equation 5.4.10 requires some further considerations. We observe that the proposals for v_s, v_{s+1} are deterministic and are given by the transformation

$$\begin{aligned}\left(v'_s, v'_{s+1}\right) = f\left(v_s, v_{s+1}\right) = \Bigg(& v_{s+1}\left(1 - v_s\right) \frac{\left[v_s + v_{s+1}\left(1 - v_s\right)\right]}{\left[v_s R_2 + v_{s+1}\left(1 - v_s\right) R_1\right]} R_1, \\ & \frac{R_2 v_s \left(v_s + v_{s+1}\left(1 - v_s\right)\right)}{v_s R_2 + v_{s+1}\left(1 - v_s\right) R_1 - v_{s+1}\left(1 - v_s\right) R_1 \left(v_s + v_{s+1}\left(1 - v_s\right)\right)}\Bigg),\end{aligned} \quad (5.4.11)$$

which is obtained from equations 5.4.1 to 5.4.7. The second ratio from equation 5.4.10 requires the Jacobian determinant of f. All its terms other than the determinant of the absolute value of the Jacobian J cancel out:

$$\begin{aligned}\frac{p\left(\boldsymbol{v}'\right)|J|}{p\left(\boldsymbol{v}\right)} &= \frac{p\left(v'_s\right)}{p\left(v_s\right)} \cdot \frac{p\left(v'_{s+1}\right)}{p\left(v_{s+1}\right)} \cdot |J| \\ &= \frac{\alpha \cancel{\left(1 - v'_s\right)^{\alpha-1}}}{\alpha \cancel{\left(1 - v_s\right)^{\alpha-1}}} \cdot \frac{\alpha \cancel{\left(1 - v'_{s+1}\right)^{\alpha-1}}}{\alpha \cancel{\left(1 - v_{s+1}\right)^{\alpha-1}}} \cdot |J| \\ &= |J|,\end{aligned}$$

where we motivate the cancellation by proposition 2 in Hastie et al., 2015, according to which

$$\left(1 - v'_s\right) \cdot \left(1 - v'_{s+1}\right) = \left(1 - v_s\right) \cdot \left(1 - v_{s+1}\right).$$

The Jacobian of f can be derived from equation 5.4.11:

$$J = \frac{R_1 R_2 \left(v_s - 1\right) \left(v_s v_{s+1} - v_s - v_{s+1}\right)^2}{\left(R_1 v_{s+1}\left(v_s - 1\right) - R_2 v_s\right) \left[R_1 \left(v_s - 1\right)^2 v_{s+1}\left(v_{s+1} - 1\right) - R_2 v_s\right]}.$$

For the sake of precision, although we did not explicitly invoke the Jacobian with the

previous moves in sections 5.2 and 5.3, or with any other of the parameters affected by move 3, nor do the original articles where move 1 and 2 were introduced, it does not mean that these moves do not require the Jacobian determinant to be taken into consideration; its absolute value just happened to be 1 in previous sections, hence it could be omitted for brevity. While deterministic proposals for Metropolis jumps are unusual, it is known that they do entail an evaluation of the Jacobian (Hastie & Green, 2012, page 314).

The acceptance ratio of move 3 is therefore:

$$\begin{aligned} R &= \min\left\{1, \gamma \cdot \frac{\cancel{p\left(\boldsymbol{y} \mid \boldsymbol{r}', \boldsymbol{m}'\right)}}{\cancel{p\left(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{m}\right)}} \cdot \frac{\cancel{p\left(\boldsymbol{m}'\right)}}{\cancel{p\left(\boldsymbol{m}\right)}} \cdot \frac{p\left(\boldsymbol{r}' \mid \boldsymbol{v}'\right)}{p\left(\boldsymbol{r} \mid \boldsymbol{v}\right)} \cdot \frac{p\left(\boldsymbol{v}'\right)}{p\left(\boldsymbol{v}\right)}\right\} \\ &= \min\left\{1, \gamma \cdot \frac{p\left(\boldsymbol{r}' \mid \boldsymbol{v}'\right)}{p\left(\boldsymbol{r} \mid \boldsymbol{v}\right)} \cdot \frac{p\left(\boldsymbol{v}'\right)}{p\left(\boldsymbol{v}\right)}\right\} \\ &= \min\left\{1, \gamma \cdot |J| \cdot \left(\frac{w_s + w_{s+1}}{w_{s+1}R_1 + w_s R_2}\right)^{n_s + n_{s+1}} R_2^{n_s} R_1^{n_{s+1}}\right\}, \end{aligned} \tag{5.4.12}$$

where γ is set as we outline in Section 5.3.2.

In section 5.7 we test the move with and without our amendment to the acceptance ratio.

5.5 Move 4

We observe the following limitations with move 2 and move 3:

- move 2 picks two juxtaposed clusters s and $s+1$, it switches their cluster assignment labels (in $\boldsymbol{r}$) and their locations (m_s, m_{s+1}), but instead of switching their weights (w_s, w_{s+1}), it switches (v_s, v_{s+1}). Since $p\left(r_i = h \mid \boldsymbol{w}\right) = w_h$, for every i including $i = s$ and $i = s+1$, we argue that it would be more consistent to switch (w_s, w_{s+1}) instead;
- move 3 aims to "simultaneously propose an update of the new cluster weights so they are something like their expected value conditional upon their allocations" (Hastie et al., 2015), and while this is absolutely fine, in our view the proposal

for $\left(w'_s, w'_{s+1}\right)$ is very complex (see equations 5.4.3 to 5.4.7), hence we are motivated to try out a simpler proposal, which switches (w_s, w_{s+1}) instead.

We therefore devise a new move (*move 4*), which takes inspiration from move 2 but instead of switching (v_s, v_{s+1}), it switches (w_s, w_{s+1}). By doing so, amongst $(\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{w} \mid \boldsymbol{y})$ a successful move only affects the stick-breaking cluster membership label (r_i) of each data point i impacted, and not any other of its associated parameters (contrary to move 1, 2 and 2 – see the example in Table 5.1).

move	i	r_i	w_{r_i}	m_{r_i}	r'_i	$w'_{r'_i}$	$m'_{r'_i}$	v_{r_i}	$v'_{r'_i}$
move 1	1	1	0.30	0.37	2	0.10	0.37	0.30	0.14
move 1	2	2	0.10	0.60	1	0.30	0.60	0.14	0.30
move 2	1	1	0.30	0.37	2	0.26	0.37	0.30	0.30
move 2	2	2	0.10	0.60	1	0.14	0.60	0.14	0.14
move 3	1	1	0.30	0.37	2	0.23	0.37	0.30	0.28
move 3	2	2	0.10	0.60	1	0.17	0.60	0.14	0.17
move 4	1	1	0.30	0.37	2	0.30	0.37	0.30	0.33
move 4	2	2	0.10	0.60	1	0.10	0.60	0.14	0.10

Table 5.1: Imaginary example of the successful outcome of move 1, 2, 3 and 4 on a hypothetical dataset of 2 observations indexed by $i = 1, 2$, when $\alpha = 1$, and $s = 1$ is selected as the cluster to switch.

The steps of move 4 are as follows:

- set $r_{\max} = \max_{1 \leq i \leq n} r_i$ and draw s uniformly at random from $\{1, \ldots, r_{\max}\}$, to select clusters s and $s+1$ (or alternatively select s uniformly at random from $\{1, \ldots, T\}$ as we outline in Section 5.3.2, and set $\gamma = 1$);

- switch the corresponding elements of $\boldsymbol{m}$:

$$m'_s = m_{s+1}$$

$$m'_{s+1} = m_s$$

- switch the values in $\boldsymbol{r}, \forall i \in \{1, \ldots, n\}$:

$$r_i' = \begin{cases} s+1, \text{ for } i \in \{i : r_i = s\} \\ s, \text{ for } i \in \{i : r_i = s+1\} \\ r_i, \text{ otherwise} \end{cases}$$

- switch (w_s, w_{s+1}):

$$w_s' = w_{s+1}$$
$$w_{s+1}' = w_s$$

Since dealing with the joint probability distribution of $\boldsymbol{v}$ is easier than dealing with $\boldsymbol{w}$ (as $\boldsymbol{v}$ is a vector of *independent* random variables), we rewrite the last move above in terms of $\boldsymbol{v}$. The $\boldsymbol{v}$ transform which achieves the effect of switching (w_s, w_{s+1}) is:

$$f(v_s, v_{s+1}) = \left(v_s', v_{s+1}'\right), \tag{5.5.1}$$

with

$$v_s' = (1 - v_s)\, v_{s+1}$$
$$v_{s+1}' = \frac{v_s}{1 - v_{s+1} + v_s v_{s+1}}$$

which can be easily verified to lead to:

$$w_s' = \left[\prod_{l<s} (1 - v_l)\right] (1 - v_s)\, v_{s+1} = w_{s+1}$$
$$w_{s+1}' = \left[\prod_{l<s} (1 - v_l)\right] (1 - v_s')\, \frac{v_s}{1 - v_{s+1} + v_s v_{s+1}} = w_s$$

and which can also be verified to be symmetric:

$$v_s'' = (1 - v_s')\, v_{s+1}' = \cancel{[1 - ((1 - v_s)\, v_{s+1})]}\, \cancel{\frac{v_s}{1 - v_{s+1} + v_s v_{s+1}}} = v_s$$
$$v_{s+1}'' = \frac{v_s'}{1 - v_{s+1}' + v_s' v_{s+1}'} = \frac{(1 - v_s)\, v_{s+1}}{1 - \cancel{\frac{v_s}{1 - v_{s+1} + v_s v_{s+1}}}\, \cancel{[1 - (v_{s+1} - v_s v)_{s+1})]}} = v_{s+1}$$

No elements of $\boldsymbol{w}$ other than w_s, w_{s+1} are impacted by this move, as

$$(1-v'_s)\left(1-v'_{s+1}\right) = (1-v_s)(1-v_{s+1}).$$

The full conditional of the parameters being updated is:

$$\begin{aligned} p(\boldsymbol{r},\boldsymbol{m},\boldsymbol{v} \mid \boldsymbol{y},\boldsymbol{u}) &\propto p(\boldsymbol{y} \mid \boldsymbol{r},\boldsymbol{m}) \cdot p(\boldsymbol{r},\boldsymbol{m},\boldsymbol{v} \mid \boldsymbol{u}) \\ &\propto p(\boldsymbol{y} \mid \boldsymbol{r},\boldsymbol{m}) \cdot p(\boldsymbol{u} \mid \boldsymbol{r},\boldsymbol{v}) \cdot p(\boldsymbol{m}) \cdot p(\boldsymbol{r},\boldsymbol{v}) \\ &= p(\boldsymbol{y} \mid \boldsymbol{r},\boldsymbol{m}) \cdot p(\boldsymbol{u} \mid \boldsymbol{r},\boldsymbol{v}) \cdot p(\boldsymbol{m}) \cdot p(\boldsymbol{r} \mid \boldsymbol{v}) \cdot p(\boldsymbol{v}), \end{aligned}$$

where the first 4 factors obviously cancel out, in the calculation of the acceptance ratio, leaving only:

$$R = \min\left\{1, \gamma \cdot \frac{p(\boldsymbol{v}')}{p(\boldsymbol{v})} \cdot |J|\right\}.$$

The ratio $p(\boldsymbol{v}')/p(\boldsymbol{v})$ cancels out too:

$$\begin{aligned} \frac{p(\boldsymbol{v}')}{p(\boldsymbol{v})} &= \frac{p(v'_s)\,p\left(v'_{s+1}\right)}{p(v_s)\,p(v_{s+1})} = \frac{p\left(v_{s+1}(1-v_s)\right)p\left(\frac{v_s}{1-v_{s+1}+v_sv_{s+1}}\right)}{p(v_s)\,p(v_{s+1})} \\ &= \frac{\left(1-v_{s+1}(1-v_s)\right)^{\alpha-1}\alpha}{(1-v_s)^{\alpha-1}\alpha} \frac{\left(1-\frac{v_s}{1-v_{s+1}+v_sv_{s+1}}\right)^{\alpha-1}\alpha}{(1-v_{s+1})^{\alpha-1}\alpha} \\ &= \left(\frac{\cancel{1-v_{s+1}+v_sv_{s+1}}}{1-v_s} \frac{\frac{1-v_{s+1}+v_sv_{s+1}-v_s}{\cancel{1-v_{s+1}+v_sv_{s+1}}}}{1-v_{s+1}}\right)^{\alpha-1} \\ &= \left(\frac{1-v_{s+1}+v_sv_{s+1}+1-v_s}{(1-v_s)(1-v_{s+1})}\right)^{\alpha-1} = 1. \end{aligned}$$

The Jacobian matrix of the transformation of v_s, v_{s+1} is:

$$\left[\frac{\partial f}{\partial v_s}, \frac{\partial f}{\partial v_{s+1}}\right] = \begin{bmatrix} -v_{s+1} & 1-v_s \\ \frac{1-v_{s+1}}{(1-v_{s+1}+v_sv_{s+1})^2} & \frac{v_s(1-v_s)}{(1-v_{s+1}+v_sv_{s+1})^2} \end{bmatrix},$$

and the absolute value of the Jacobian determinant of the transform (see equation 5.5.1) is:

$$\begin{aligned} |J| &= \left| -\frac{v_{s+1}v_s(1-v_s)}{(1-v_{s+1}+v_sv_{s+1})^2} - \frac{(1-v_s)(1-v_{s+1})}{(1-v_{s+1}+v_sv_{s+1})^2} \right| \\ &= \left| -\frac{(1-v_s)\cancel{(1-v_{s+1}+v_sv_{s+1})}}{(1-v_{s+1}+v_sv_{s+1})^{\cancel{2}}} \right| \end{aligned}$$

$$= \frac{1 - v_s}{1 - v_{s+1}(1 - v_s)}.$$

The acceptance ratio of move 4 is therefore given by:

$$\begin{aligned} R &= \min\left\{1, \gamma \cdot \frac{1 - v_s}{1 - v_{s+1}(1 - v_s)}\right\} \\ &= \min\left\{1, \gamma \cdot \frac{1 - w_1 - \ldots - w_s}{1 - w_1 - \ldots - w_{s-1} - w_{s+1}}\right\}, \end{aligned}$$

where γ is outlined in Section 5.3.2.

5.6 Efficiency measures

It is known that samples from MCMC chains suffer to various degrees from autocorrelation, which renders them less efficient than an equally-sized independent sample. This is well-articulated in Sokal, 1997, which discusses the concept of *integrated autocorrelation time* of an observable f, denoted by τ:

$$\tau = \frac{1}{2} \sum_{l=-\infty}^{\infty} \rho_l = \frac{1}{2} + \sum_{l=1}^{\infty} \rho_l,$$

where ρ_l is the autocorrelation of f at lag l. Measuring τ is important when comparing MCMC algorithms because 2τ measures how much larger the variance of the estimate of f is than in independent sampling. Like other studies before ours (Green & Richardson, 2001; Kalli et al., 2011; Neal, 2000; Papaspiliopoulos & Roberts, 2008), we also turn to τ to gauge the efficiency of the algorithms discussed in this chapter.

In an MCMC chain with N iterations, producing $\hat{\tau}$, the sample estimate of τ, requires the introduction of a cutoff $M \ll N$ in the summation, with "M large enough so that ρ_l is negligible for $|l| > M$"; this is due to reasons outlined in Sokal, 1997, and which in essence pertain to the fact that, otherwise, the variance of $\hat{\tau}$ would not converge to 0 for $N \to \infty$, which is undesirable behaviour for an estimator. We

therefore use the formula

$$\hat{\tau} = \frac{1}{2} + \sum_{l=1}^{M} \hat{\rho}_l,$$

where $M := \min\{l \in \mathbb{N} : l \geq 10 \cdot \hat{\tau}(l)\}$, as per the advice in Sokal, 1997, page 145. We also report in our tables in section 5.7 the value of the standard deviation of the estimator, calculated as:

$$\sqrt{\frac{2(2M+1)}{n}}\ \hat{\tau}.$$

The functionals that we turn to, to measure the integrated autocorrelation time, are:

- K_n, the number of clusters. We choose this metric for comparability, as it has been widely employed in previous studies. Although K_n is not directly altered by a successful label switch, its integrated autocorrelation time should still be informative of the overall convergence speed;

- the deviance as defined in Green and Richardson, 2001; Kalli et al., 2011; Neal, 2000; Papaspiliopoulos and Roberts, 2008:

  $$D = -2\sum_{i=1}^{n} \log\left\{\sum_{j} \frac{n_j}{n} p(y_i \mid \theta_j)\right\}, \tag{5.6.1}$$

  Previous studies justify its use on the basis that it is seen as "a global function of all model parameters" (Kalli et al., 2011);

- θ_1, which was introduced in Neal, 2000. Theoretically, since our testing data set is composed of 320 observations constituted of only 9 unique values, one could monitor θ_i for nine choices of i, corresponding to each unique value of $\boldsymbol{y}$, however for brevity we only report information about θ_1;

- r_1, which was never considered in previous studies but which in our view is important, since it is *label switching* which moves 1, 2, 3 and 4 all attempt to accelerate;

- w_1 and m_1 too, for completeness.

5.7 Testing

We present in this section some results pertaining to moves 1, 2, 3, 4, and to two further variations:

- move $2^\star$ stands for move 2 without the two Metropolis-Hastings adjustments that we describe in section 5.3.2; in fact, move $2^\star$ is only attempted for $k > 1$, and it works by selecting s from $\{1, \dots, \max(\boldsymbol{r}) - 1\}$;
- similarly, move $3^\star$ is also only attempted for $k > 1$, and it works by selecting s from $\{1, \dots, \max(\boldsymbol{r}) - 1\}$, which is how move 3 is described in Hastie et al., 2015. Furthermore, move $3^\star$ uses the acceptance rate formula from the same article (see equation 5.4.8), rather than our complete acceptance rate formula (see equation 5.4.12).

The data set that we use is the thumb tack data set of Beckett and Diaconis, 1994, as it appears in Liu, 1996. The data set is composed of 320 observations, pertaining to the roll of a thumb tack; each tack was flipped 9 times, and "a one was recorded if the tack landed point up". We model it via a Dirichlet process mixture with a binomial likelihood, and we set G_0 to a Beta$(1, 1)$. For testing purposes, we prefer to fix $\alpha = 1$ as opposed to assigning a prior on it, to make it easier to appraise convergence; calculations for $\alpha = 0.2$ and $\alpha = 5$ (not included here for brevity) broadly lead to the same conclusions, and the methods we outline in this chapter remain applicable even when α is random. We use native R (R Core Team, 2022) to perform $2,000,000$ iterations of the slice sampler.

As expected, empirical tests confirm that the slice sampler with move $2^\star$ does not converge to the right posterior; this can be clearly seen in figure 5.1, where we compare its mean posterior of $K \mid \boldsymbol{y}$ with the one from collapsed algorithm 2. The two can be clearly seen to take close but very distinct paths, and to persist on those paths over the entire spectrum of iterations. Since collapsed algorithm 2 is known to be very stable, since all other algorithms[2] converge to values very close to where

[2]Except for the slice sampler with move $3^\star$.

collapsed algorithm 2 does, and most importantly since theory also indicates so, we conclude that the convergence issue lies with move 2* and not with any of the other samplers.

Move 3* is also affected by lack of convergence of the mean of the posterior of $K \mid \boldsymbol{y}$ to the correct value; in fact, it converges to a value approximately equal to 9 – we do not attempt to display it in figure 5.1 as it would be out of scale. We also observe that in our tests move 3* returns an empirical acceptance rate close to 0.95, whereas the one that we obtain in our tests with its improved version ('move 3') is approximately 0.62 (see table 5.2).

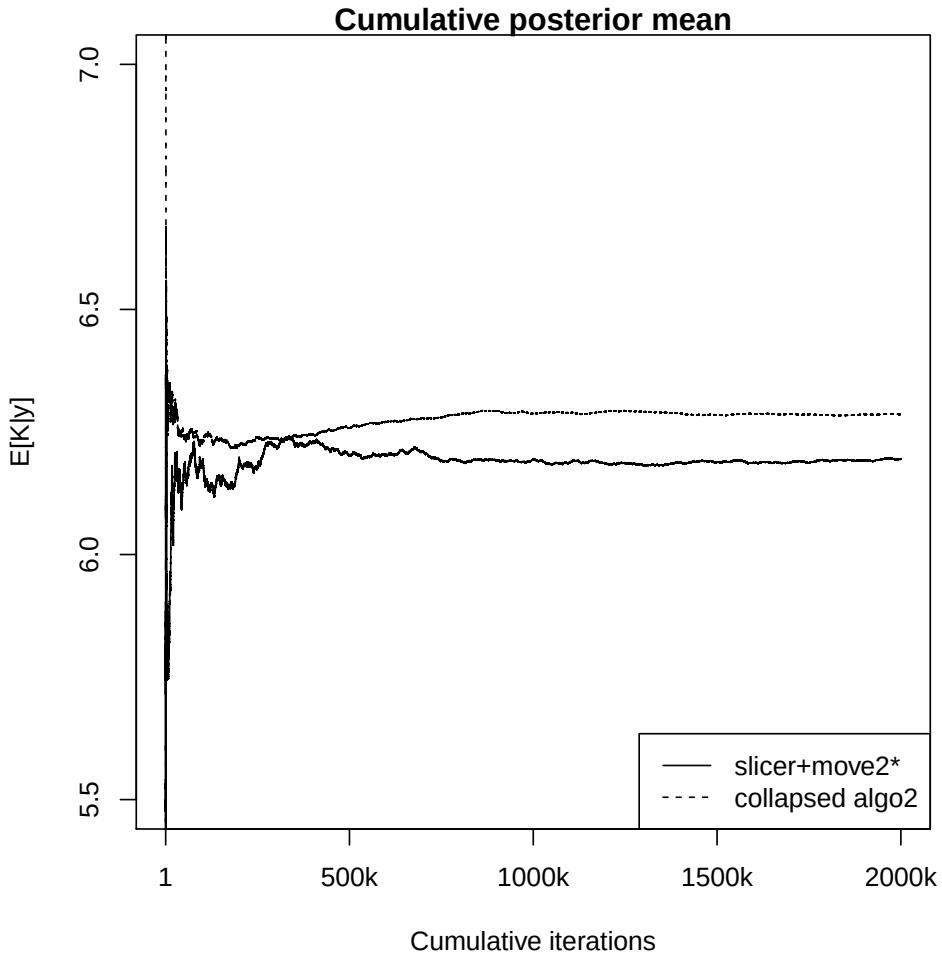

Figure 5.1: Impact of an incorrect Metropolis jump that overlooks the adjustments described in 5.3.2, when comparing against collapsed algorithm 2 (used here as a reference), over $2,000,000$ iterations and for $\alpha = 1$.

Move	Average AR
1	0.09
2	0.27
3	0.62
4	0.59
$2^\star$	0.25
$3^\star$	0.95

Move	IAT_K	IAT_{w_1}	IAT_{r_1}	$\text{IAT}_{w_{r_1}}$	IAT_{m_1}	IAT_{θ_1}	IAT_D
none	75.16 (2.92)	126.00 (6.33)	43.70 (1.29)	36.00 (0.97)	388.12 (34.20)	0.87 (0.004)	6.43 (0.07)
1	72.16 (2.74)	121.15 (5.97)	32.15 (0.82)	34.34 (0.90)	38.69 (1.08)	0.87 (0.004)	6.65 (0.08)
2	65.39 (2.37)	73.59 (2.82)	32.96 (0.85)	32.44 (0.83)	226.31 (15.23)	0.86 (0.004)	6.39 (0.07)
3	57.37 (1.94)	34.38 (0.90)	13.64 (0.23)	29.63 (0.72)	6.15 (0.07)	0.86 (0.004)	6.24 (0.07)
4	59.96 (2.08)	35.02 (0.93)	13.68 (0.23)	29.08 (0.70)	6.33 (0.07)	0.85 (0.004)	6.29 (0.07)

Table 5.2: Average empirical acceptance rates of moves 1, 2, 3, 4, $2^\star$ and $3^\star$, and integrated autocorrelation time of moves 1, 2, 3, 4, over $2,000,000$ iterations with $\alpha = 1$. Estimate of the standard errors are in parenthesis.

5.8 Conclusions

Along all the integrated autocorrelation times considered in table 5.2, move 3[3] and move 4 stand out as the most efficient. Although move 3 has a slight advantage over most of the measures included in table 5.2, the standard deviation of its IAT estimators suggests possible reversals in the positions of move 3 and move 4 due to randomness, even after 2 million iterations; this is due to the fact that they are close in performance, which makes it more difficult to precisely rank them. Further limitations of table 5.2 are that:

- the estimation of the integrated autocorrelation time is inherently imprecise, for the reasons outlined in Sokal, 1997;
- the outcomes are specific to the data set we tested;

hence the table is only intended to give an indication of relative performance, rather than to be precise.

[3]Specifically, our improved version of move 3, as per what we discuss in this chapter.

Although the outcomes reported in table 5.2 are specific to $\alpha = 1$, in our experiments we observed similar ordering for other fixed values of α too, on the same data set.

Overall, there does not appear to be a performance advantage to be gained from move 4 over move 3, although move 4 could still be preferred over move 3 where simplicity and clarity of the proposal mechanism is important, notwithstanding indications of marginally better performance from move 3.

We conclude by anticipating that, anyway, in chapters 6 and 7 we introduce approaches which entirely dominate the gains in performance arising from any of these Metropolis jumps.

Chapter 6

Sequential importance sampling for stick-breaking

To accelerate the estimation of the stick-breaking parameters $\boldsymbol{r}$ and $\boldsymbol{w}$, and to improve on the relatively long integrated autocorrelation times of the MCMC chains produced by Gibbs-type conditional samplers, we turn to sequential importance sampling, which as a general method is known to produce i.i.d. observations and which therefore should be free from the aforementioned issues.

In the DPM context, sequential importance sampling was first explored by Liu, 1996, who introduced a sampler for $\boldsymbol{s}, \boldsymbol{\theta} \mid \boldsymbol{y}$ (algorithm S1); S. N. MacEachern et al., 1999 later introduced a variation (algorithm S2) which integrates $\boldsymbol{\theta}$ out and directly aims to estimate $\boldsymbol{s} \mid \boldsymbol{y}$, with higher efficiency; S2 was also used in Quintana, 1998.

As $\boldsymbol{s}, \boldsymbol{\theta} \mid \boldsymbol{y}$ are parameters from the collapsed Polya urn representation of the DPM, while we are interested in the parameters $\boldsymbol{r}, \boldsymbol{w} \mid \boldsymbol{y}$ from the stick-breaking representation, we develop a new sequential importance sampling for $\boldsymbol{r} \mid \boldsymbol{y}$ (algorithm R), which operates in the stick-breaking space and which assigns cluster membership labels ($\boldsymbol{r}$) pointing to which specific stick in the stick-breaking construction process each observation arises from. We then complete it with a blocking scheme which allows to sample $\boldsymbol{w}, \boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{r}$ too (the length and location of the sticks), while still producing i.i.d. observations.

6.1 Sequential importance sampling

According to Robert and Casella, 2004, section 3.3, importance sampling moves from the observation that the integral in equation 6.1.1 can be numerically approximated via an appropriately weighted sum of samples drawn from the importance sampling distribution g, rather than directly drawn from the target distribution f:

$$\begin{aligned}\mathbb{E}_f\left[h\left(X\right)\right] &= \int_{\chi} h\left(x\right) f\left(x\right) \mathrm{d}x = \int_{\chi} h\left(x\right) \frac{f\left(x\right)}{g\left(x\right)} g\left(x\right) \mathrm{d}x \\ &\approx \frac{1}{B} \sum_{b=1}^{B} h\left(X^{(b)}\right) \frac{f\left(X^{(b)}\right)}{g\left(X^{(b)}\right)} \\ &= \frac{1}{B} \sum_{b=1}^{B} h\left(X^{(b)}\right) W_b.\end{aligned} \tag{6.1.1}$$

When the normalising constant of f is unknown, the weights W_b are often re-scaled as follows, to have mean 1:

$$W_b^{\star} = \frac{BW_b}{\sum_{b=1}^{B} W_b}. \tag{6.1.2}$$

For a more comprehensive discussion of its properties we refer the reader to the reference above.

Sequential importance sampling (SIS) is to all effects an *importance sampling* method; it moves from the same idea, and its name only emphasizes the involvement of a joint distribution which is obtained in turns, by sampling from various conditionals, in a sequence.

Assume that the objective is to sample from the posterior distribution $p\left(\boldsymbol{\xi} \mid \boldsymbol{y}\right)$, where $\boldsymbol{\xi}$ is the parameter vector and $\boldsymbol{y}$ is the data. According to Liu, 1996, sequential importance sampling entails drawing from

$$p\left(\xi_i \mid y_1, \xi_1, \ldots, y_{i-1}, \xi_{i-1}, y_i\right), \quad i = 1, \ldots, n, \tag{6.1.3}$$

so that the joint importance sampling distribution is

$$p\left(\xi_1 \mid y_1\right) \prod_{i=2}^{n} p\left(\xi_i \mid y_1, \xi_1, \ldots, y_{i-1}, \xi_{i-1}, y_i\right).$$

The unnormalised importance weights are given by the ratio between the target

distribution $p(\boldsymbol{\xi} \mid \boldsymbol{y})$ and the importance sampling distribution:

$$\begin{aligned} W &= p(\boldsymbol{\xi} \mid \boldsymbol{y}) \frac{1}{p(\xi_1 \mid y_1) \prod_{i=2}^{n} p(\xi_i \mid y_1, \xi_1, \ldots, y_{i-1}, \xi_{i-1}, y_i)} \\ &= \frac{\cancel{p(\boldsymbol{\xi}, \boldsymbol{y})}}{p(\boldsymbol{y})} \frac{p(y_1)}{p(\xi_1, y_1)} \frac{p(\xi_1, y_1, y_2)}{p(\xi_1, \xi_2, y_1, y_2)} \cdots \frac{p(\xi_1, \ldots, \xi_{n-1}, y_1, \ldots, y_n)}{\cancel{p(\xi_1, \ldots, \xi_n, y_1, \ldots, y_n)}} \\ &= \frac{p(y_1)}{p(\boldsymbol{y})} \prod_{i=2}^{n} p(y_i \mid \xi_1, y_1, \ldots, \xi_{i-1}, y_{i-1}) \\ &\propto \prod_{i=2}^{n} p(y_i \mid \xi_1, y_1, \ldots, \xi_{i-1}, y_{i-1}), \end{aligned} \tag{6.1.4}$$

while the normalised weights are given by equation 6.1.2.

6.2 Algorithm S1

Liu, 1996 used sequential importance sampling to estimate the parameter vector $\boldsymbol{\theta}$ in the context of a DPM with $G_0 \sim \text{Beta}(a, b)$ and with a binomial likelihood $p(y_i \mid \theta_i; l_i)$, where l_i is the number of trials associated with each observation.

Equation 6.1.3 in this modelling context leads to

$$\begin{aligned} &p(\theta_i \mid \theta_1, \ldots, \theta_{i-1}, y_1, \ldots, y_{i-1}, y_i) \propto p(y_i, \theta_i \mid \theta_1, \ldots, \theta_{i-1}, y_1, \ldots, y_{i-1}) \\ &= p(\theta_i \mid \theta_1, \ldots, \theta_{i-1}, y_1, \ldots, y_{i-1}) \cdot p(y_i \mid \theta_1, \ldots, \theta_{i-1}, \theta_i, y_1, \ldots, y_{i-1}) \\ &= p(\theta_i \mid \theta_1, \ldots, \theta_{i-1}) \cdot p(y_i \mid \theta_1, \ldots, \theta_{i-1}, \theta_i) \\ &\propto \begin{cases} \frac{\alpha}{\alpha+i-1} \binom{l_i}{y_i} \frac{B(a+y_i, b+l_i-y_i)}{B(a,b)} & \text{if } \theta_i \notin \{\theta_1, \ldots, \theta_{i-1}\} \\ \frac{\left[\sum_{l=1}^{i-1} \delta_{\theta_j}(\theta_l)\right]}{\alpha+i-1} \binom{l_i}{y_i} \theta_j^{y_i} (1-\theta_j)^{l_i-y_i} & \text{if } \theta_i = \theta_j, j < i \end{cases} \end{aligned}$$

which is a mixture of distributions – it is in fact the same equation as 3.3.7, which we outlined in the context of collapsed Algorithm 1 (see section 3.3.3). The weights are obtained from equation 6.1.4.

Since repeated values in the location vector parameter identify membership to clusters, estimating $\boldsymbol{\theta}$ implicitly allows to infer $\boldsymbol{s}$ too (S. N. MacEachern et al., 1999).

6.3 Algorithm S2

Just like SIS algorithm S1 repurposes the same conditional used in the collapsed Gibbs Algorithm 1 to the sequential importance sampling setting, so does SIS algorithm S2 with collapsed Gibbs algorithm 2 (see section 3.3.4 and equation 3.3.10). Instead of sampling $\boldsymbol{\theta} \mid \boldsymbol{y}$, algorithm S2 produces $\boldsymbol{s} \mid \boldsymbol{y}$.

Equation 6.1.3 in this modelling context leads to

$$
\begin{aligned}
&p\left(s_i \mid s_1, \ldots, s_{i-1}, y_1, \ldots, y_{i-1}, y_i\right) \propto p\left(y_i, s_i \mid s_1, \ldots, s_{i-1}, y_1, \ldots, y_{i-1}\right) \\
&= p\left(s_i \mid s_1, \ldots, s_{i-1}, y_1, \ldots, y_{i-1}\right) \cdot p\left(y_i \mid s_1, \ldots, s_{i-1}, s_i, y_1, \ldots, y_{i-1}\right) \\
&= p\left(s_i \mid s_1, \ldots, s_{i-1}\right) \cdot p\left(y_i \mid s_1, \ldots, s_{i-1}, s_i\right) \\
&\propto \begin{cases} \frac{\alpha}{\alpha+i-1}\binom{l_i}{y_i}\frac{B(a+y_i, b+l_i-y_i)}{B(a,b)} & \text{if } s_i \notin \{s_1, \ldots, s_{i-1}\} \\ \frac{\left[\sum_{l=1}^{i-1}\delta_{s_j}(s_l)\right]}{\alpha+i-1}\binom{l_i}{y_i}\frac{B\left(a+y_i\sum_{\{z:r_z=i,z<i\}}, b+l_i-y_i+\sum_{\{z:r_z=i,z<i\}}\right)}{B\left(a+\sum_{\{z:r_z=i,z<i\}} y_z, b+\sum_{\{z:r_z=i,z<i\}} l_z-y_z\right)} & \text{if } s_i = s_j, j<i \end{cases}
\end{aligned}
$$

The weights are obtained from equation 6.1.4.

S. N. MacEachern et al., 1999 demonstrated how algorithm S2 is a Rao-Blackwellisation of algorithm S1 and therefore is guaranteed to be more efficient than S1.

6.4 Algorithm R

We observe that one of the main advantages of sequential importance sampling over Gibbs sampling is that it returns an i.i.d. sample, hence this methodology seems a good candidate to solve the issue of stick-breaking samplers having trouble exploring all of their multiple local modes. We note that the sequential importance sampler was last used in the DPM context in 1996 (Liu, 1996) and 1999 (S. N. MacEachern et al., 1999) respectively, before stick-breaking samplers were widely introduced. It seems therefore a good idea to attempt and formulate a sequential importance sampler specifically designed for stick-breaking. To make the sampler as effective as possible, we design it to purely target $\boldsymbol{r} \mid \boldsymbol{y}$; blocking strategies will allow to infer

all other remaining parameters from that.

As we will see in this section, just like SIS algorithm 1 and 2, SIS algorithm R too works best for conjugate pairs. As outlined in section 3.3.2, lack of conjugacy does not invalidate these algorithms, however it can make them considerably less attractive due to higher computational costs. Because our test data happens to naturally fit the beta/binomial pair[1], which is conjugate, we do not explore sequential importance samplers for non-conjugate pairs in this chapter; for the non-conjugate case, we refer to the methods that we develop in chapter 7, which are equally well suited to conjugate and non-conjugate pairs. For completeness, we also observe that, just like a parallel exists between SIS algorithms 1 and 2 and collapsed algorithms 1 and 2 (as they essentially rely on the same posterior), so can SIS algorithm R be repurposed to be a Gibbs collapsed algorithm by a simple adaptation of equation 6.4.1; even so, the Gibbs adaptation of SIS algorithm R continues to be best suited for conjugate pairs.

We wish to draw sequentially from

$$p\left(r_i \mid y_1, r_1, \ldots, y_{i-1}, r_{i-1}, y_i\right) = \frac{p\left(y_1, \ldots, y_i \mid r_1, \ldots, r_i\right) \cdot p\left(r_i \mid r_1, \ldots, r_{i-1}\right)}{p\left(y_1, \ldots, y_i \mid r_1, \ldots, r_{i-1}\right)}. \tag{6.4.1}$$

To do so, we first draw $\nu_i \sim \text{Unif}\,(0,1)$, then we incrementally calculate $p\left(r_i = h \mid \ldots\right)$ with equation 6.4.1, for $h = 1, 2, \ldots$, until the cumulative probability exceeds ν_i, to determine our draw. All that remains is therefore to show how to calculate the expression in equation 6.4.1.

The first factor at the numerator of equation 6.4.1 is easy to calculate as our case is conjugate:

$$p\left(y_1, \ldots, y_i \mid r_1, \ldots, r_i\right) = \left[\prod_{j=1}^{i} \binom{n_j}{y_j}\right] \cdot \left(\prod_{c \in R_i^\star} \frac{B\left(a + \sum_{\{l:r_l=c\}} y_l, b + \left(\sum_{\{l:r_l=c\}} n_l - y_l\right)\right)}{B\left(a, b\right)}\right)$$

where $R_i^\star$ is the set of unique values in $(r_1, \ldots, r_i)$ and B is the beta function.

[1] We use a binomial likelihood, and we set G_0 to a Beta $(1,1)$ distribution.

The second factor at the numerator of equation 6.4.1 is also easy:

$$p\left(r_i \mid r_1, \ldots, r_{i-1}\right) = \frac{n_{r_i}}{\alpha + i - 1} \prod_{k=1}^{r_i} \frac{g_k + \alpha - 1}{g_k + \alpha},$$

where $g_k = \#\left\{i : r_i >= k\right\}$ (Miller, 2019).

The denominator of equation 6.4.1 can be obtained by observing that, while the missing cluster membership indicator r_i can assume an infinity of values in $\mathbb{N}^+$, it only influences the likelihood through a finite set of states - namely, by either being equal to any of $(r_1, \ldots, r_{i-1})$, or by being different from them all (and if so, its specific value is uninfluential). Therefore:

$$\begin{aligned} p\left(y_1, \ldots, y_i \mid r_1, \ldots, r_{i-1}\right) = \\ \left[\sum_{h \in R_i^{\star -}} p\left(y_1, \ldots, y_i \mid r_1, \ldots, r_{i-1}, r_i = h\right) \cdot p\left(r_i = h \mid r_1, \ldots, r_{i-1}\right)\right] + \\ + p\left(r_i \notin R_i^{\star -} \mid r_1, \ldots, r_{i-1}\right) \cdot p\left(y_i\right) \cdot p\left(y_1, \ldots, y_{i-1} \mid r_1, \ldots, r_{i-1}\right), \end{aligned} \tag{6.4.2}$$

where $R_i^{\star -}$ is the set of unique values of $(r_1, \ldots, r_{i-1})$.

By drawing sequentially from 6.4.1, the joint importance sampling distribution is

$$p\left(r_1 \mid y_1\right) \prod_{i=2}^{n} p\left(r_i \mid y_1, r_1, \ldots, y_{i-1}, r_{i-1}, y_i\right),$$

hence, as implied by equation 6.1.4, the unnormalised weights can be obtained as

$$\begin{aligned} W &= \frac{p\left(y_1\right)}{p\left(\boldsymbol{y}\right)} \prod_{i=2}^{n} p\left(y_i \mid r_1, y_1, \ldots, r_{i-1}, y_{i-1}\right) \\ &\propto \prod_{i=2}^{n} p\left(y_i \mid r_1, y_1, \ldots, r_{i-1}, y_{i-1}\right) \\ &= \prod_{i=2}^{n} \frac{p\left(y_1, \ldots, y_i \mid r_1, \ldots, r_{i-1}\right)}{p\left(y_1, \ldots, y_{i-1} \mid r_1, \ldots, r_{i-1}\right)}, \end{aligned}$$

where the numerator was already calculated in equation 6.4.2 (it is the inverse of the normalising constant), and the denominator was too.

6.4.1 Remaining parameters

For the remaining parameters. we adopt a blocking strategy. The full joint conditional of interest can be factored as follows:

$$\begin{aligned} p(\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{v} \mid \boldsymbol{y}) &= p(\boldsymbol{r} \mid \boldsymbol{y})\, p(\boldsymbol{m}, \boldsymbol{v} \mid \boldsymbol{r}, \boldsymbol{y}) \\ &\propto p(\boldsymbol{r} \mid \boldsymbol{y})\, p(\boldsymbol{y} \mid \boldsymbol{m}, \boldsymbol{r})\, p(\boldsymbol{m})\, p(\boldsymbol{v} \mid \boldsymbol{r}), \end{aligned} \tag{6.4.3}$$

and therefore the full algorithm can be constructed as follows:

- sample from $p(\boldsymbol{r} \mid \boldsymbol{y})$ via sequential importance sampling, as outlined in this chapter, and derive the normalised weight W_b of each sample;
- sample from $p(m_h) \cdot p(\boldsymbol{y} \mid m_h, \boldsymbol{r})$, which is proportional to the density of a Beta $\left(a + \sum_{\{i:r_i=h\}} y_i, b + \sum_{\{i:r_i=h\}} l_i - y_i\right)$, for h corresponding to all occupied clusters (and any others which are of interest);
- sample from $p(\boldsymbol{v} \mid \boldsymbol{r})$, via equation 3.4.2.

6.5 Assessment

In this section we test algorithm R against S1 and S2 on the same data set that we used in chapter 5 to gauge the performance of various Metropolis moves in the context of the slicer sampler. To run algorithm S1, S2 and R we use $N = 2,000,000$ draws each.

Integrated autocorrelation time, which we relied upon to compare Gibbs samplers against each other in chapter 5, is not the right efficiency measure for sequential importance samplers, which generate i.i.d. observations. While we still produce it in some of the following performance comparison tables, the reason why we display it is to re-iterate the point that the IAT from SIS samplers is minimal, especially when compared to the slice sampler.

To compare between SIS algorithms, we use the variance of their normalised weights, and the Effective Sample Size (ESS), calculated as follows:

$$ESS \approx \frac{N}{1 + \mathbb{V}\text{ar}\left(\hat{\boldsymbol{W}}\right)},$$

where the weight estimates $\hat{W}$ are standardised to sum to 1, as follows:

$$\hat{W}_i = \frac{W_i N}{\sum_{i=1}^{N} W_i},$$

where N is the size of the simulation sample. For more information pertaining to this performance measure, we refer to Liu, 1996; S. N. MacEachern et al., 1999.

The outcome of our tests is summarised in table 6.1 and in figure 6.1: algorithms S2 and R have a much lower variance than algorithm S1, due to the fact that they integrate $\boldsymbol{\theta}$ out, with S2 having slightly lower variance than R, as is to be expected since $\boldsymbol{r}$ has a wider parameter space than $\boldsymbol{s}$ (intuitively, with R any data point y_i can theoretically be associated with any value of $r_i \in \mathbb{N}^+$, whereas with S2 we can only have y_i associated with $s_i \in \mathbb{N}_i$). Our results for S1 and S2 are unsurprising, and confirm those of S. N. MacEachern et al., 1999; those pertaining to R show that the negative impact of the parameter space of $\boldsymbol{r} \mid \boldsymbol{y}$ being wider than $\boldsymbol{s} \mid \boldsymbol{y}$ is limited and not such to disrupt algorithm R, at least on our test data set.

All three cases in figure 6.1 show well-behaved variance, with no instability that would suggest it being infinite.

One disadvantage of all three SIS algorithms, though, is that they depend on the order of the observations in the data set, in a way that can be material (the order we used in our tests is the one from Liu, 1996). This was already apparent upon close inspection in Table 2 and Table 3 of S. N. MacEachern et al., 1999: for example, for S1, when $\alpha = 1$, the effective sample size in a sample of $10,000$ was found to be 227, while on a different run on a permutation of the data points the same decreased to 23; for S2 the same were found to be 814 and $1,091$ respectively. However, we also argue that the instability of the results in S. N. MacEachern et al., 1999 may be

largely due to the impact of Monte Carlo noise, as their tests were carried out on $10,000$ simulations only (likely due to limitations in computing power back in 1999). Our results (see figure 6.1), which employ $2,000,000$ iterations, shows less erratic behaviour of algorithm S1.

However, our results confirm the potentially large impact that ordering may have on the overall efficiency of the algorithm: for example, in an experiment we ran with $2,000,000$ iterations, algorithm R run on the thumb tack data set as ordered in Liu, 1996 exhibited a variance equal to 14.13, however the same algorithm run on two other random permutations of the data set returned a variance of 8.1 and 4.3.

In table 6.1 we display a summary of the performance measures pertaining to these three SIS algorithms, and in table 6.2 we report an example of the effect of weighting $p(r_1 = h \mid \boldsymbol{y})$, along with a comparison with the same probabilities as obtained via the slice sampler with move 4.

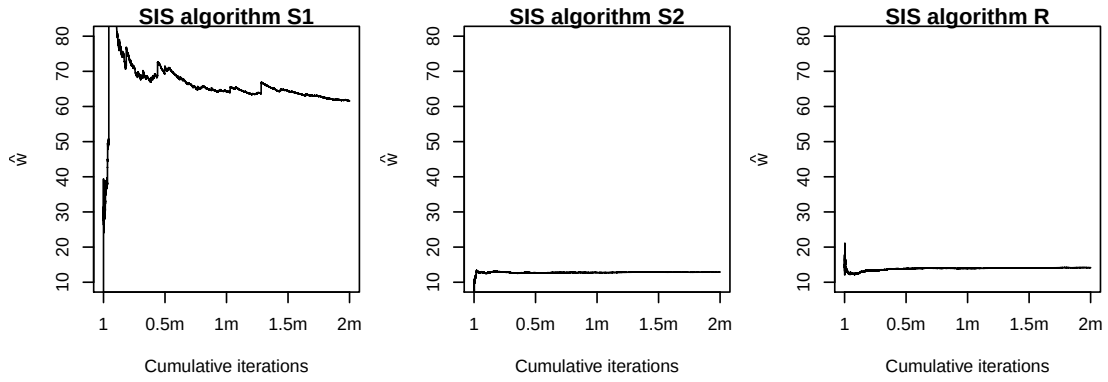

Figure 6.1: Cumulative variance of $\hat{W}$ over $2,000,000$ iterations, when using sequential importance sampling algorithms S1, S2, R on the thumb tack data set, with $\alpha = 1$.

6.6 Conclusions

While relatively more complex than S1 and S2, due to the need to compute a normalising constant which is unnecessary in the first two, algorithm R is still fairly

Algorithm	$\mathbb{V}\mathrm{ar}\left(\hat{W}\right)$	ESS	IAT_K	IAT_{r_1}	IAT_{w_1}	IAT_{m_1}
S1	61.59	31,956	0.50	NA	NA	0.50
S2	12.90	143,927	0.50	NA	NA	NA
R	14.13	132,154	0.50	0.50	0.50	0.50

Table 6.1: Comparison of sequential importance sampling algorithms S1, S2 and R, over $2,000,000$ iterations, on the thumb tack data set, with $\alpha = 1$.

	$p\left(r_1 = h \mid \boldsymbol{y}\right)$		
h	R unweighted	R weighted	slice4
1	0.5001	0.3850	0.3837
2	0.2498	0.3209	0.3201
3	0.1250	0.1670	0.1676
4	0.0626	0.0738	0.0747
5	0.0313	0.0310	0.0313
6	0.0156	0.0125	0.0126
7	0.0078	0.0055	0.0055
8	0.0039	0.0024	0.0024
9	0.0019	0.0010	0.0011
10	0.0010	0.0004	0.0005
...	...	...	...

Table 6.2: Posterior distribution of r_1, over $2,000,000$ iterations, on the thumb tack data set, with $\alpha = 1$, obtained with the SIS R algorithm, and with the slice sampler and move 4).

straightforward to implement for conjugate families.

Given the outcomes of our tests on the thumb tack dataset of 320 observations from Beckett and Diaconis, 1994, we conclude that algorithm R is very competitively placed against Gibbs-type stick-breaking samplers, as it outperforms them. By definition, being i.i.d. it is not affected by autocorrelation issues, which instead is the main problem with Gibbs samplers. Furthermore it can be parallelised with minimal effort, which is also a material advantage given today's ample availability of cloud computing solutions. The variance of its importance weights clearly needs to be monitored, just like an MCMC chain needs to be monitored for convergence, however on our test data set of 320 observations, it behaved well, reaching stability after a couple of dozen thousand iterations, with no signs of inappropriately large

weights materialising later in the chain, to materially shift the cumulative results. Algorithm R is well suited to be used either as the main sampler, or in combination or as a benchmark to the Gibbs sampler. Potentially, individual samples from it could also be used as proposals for Metropolis-within-Gibbs jumps. In our view, the main disadvantage of sequential importance sampling is that its performance depends on the order of the observations in the data set (S. N. MacEachern et al., 1999); in the absence of more conclusive studies pertaining to the optimal ordering, or to more sophisticated adjustments based on particle filtering (Fearnhead, 2004) or Sequential Monte Carlo (Ulker et al., 2011), which we have not explored, we suggest experimentation with various permutations of the observations in the data set.

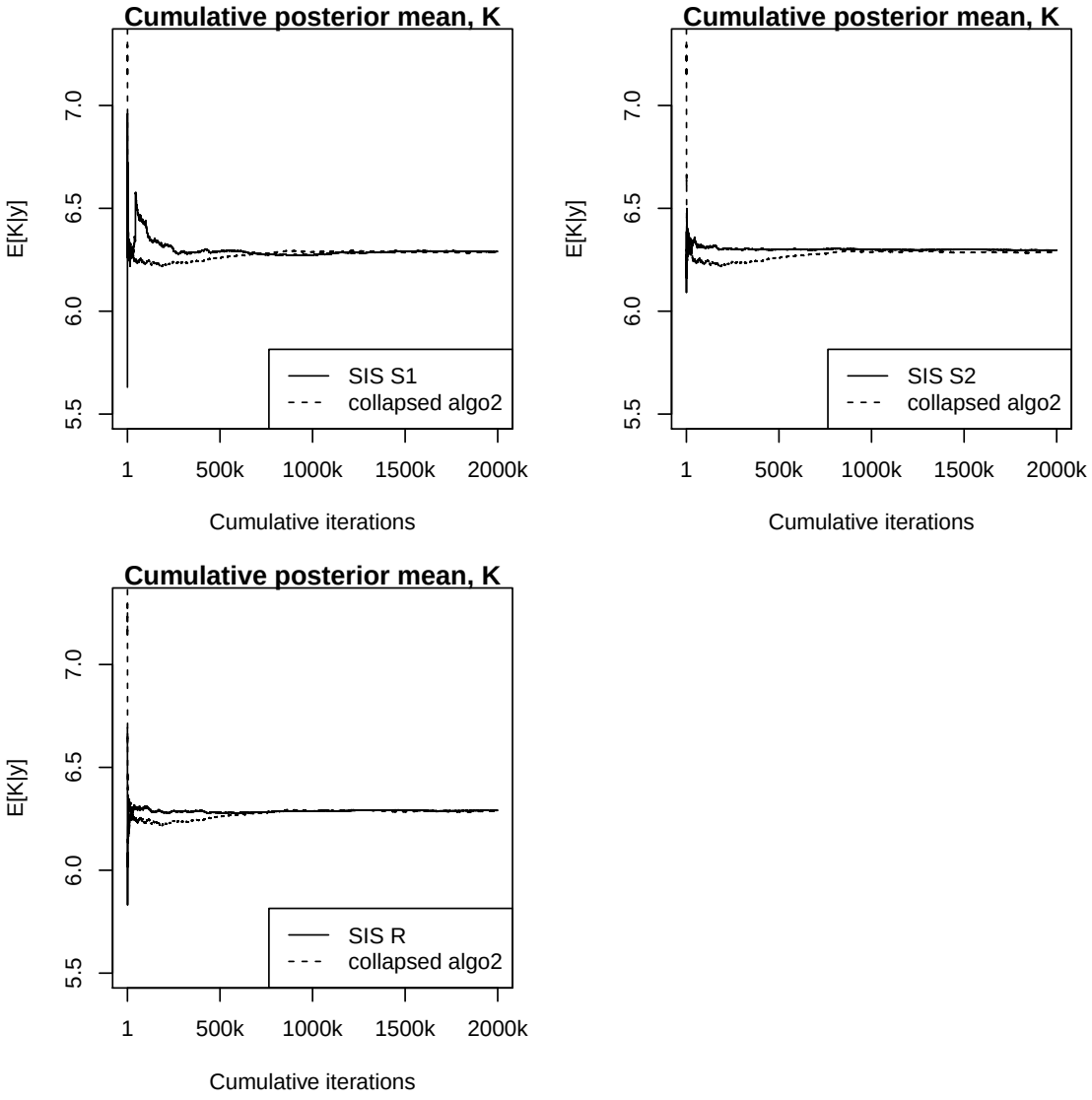

Figure 6.2: Cumulative posterior mean of K_n over $2,000,000$ iterations (calculated every 10 data points), when using sequential importance sampling algorithm S1, S2 and R, on the thumb tack data set with $\alpha = 1$, compared to collapsed algorithm 2.

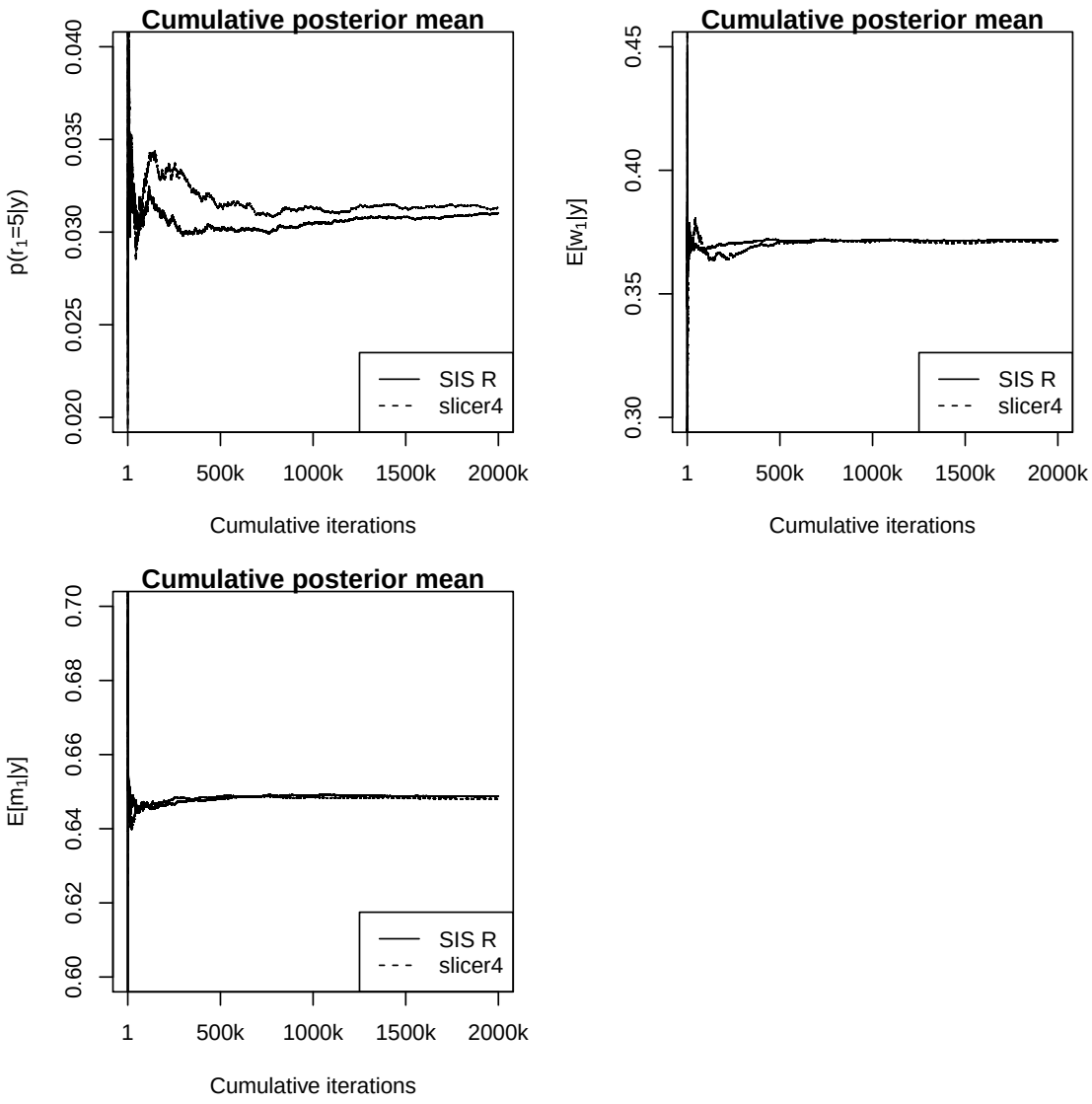

Figure 6.3: Cumulative posterior mean of $p(r_1 = 5 \mid \boldsymbol{y})$, $\mathbb{E}[w_1 \mid \boldsymbol{y}]$ and $\mathbb{E}[m_1 \mid \boldsymbol{y}]$ over $2,000,000$ iterations, when using sequential importance sampling algorithm R, on the thumb tack data set with $\alpha = 1$, compared to slice sampler algorithm with move 4.

Chapter 7

The transcoding sampler

In statistics, information about data and parameters pertaining to a model must be encoded into standard format according to some convention, for it to be useable. While doing so is often straightforward, the numerous alternate representations of the Dirichlet process offer a multiplicity of options to choose from, the relationships between which are complex, especially when it pertains to identifying partitions. This is partly inherited from the fact that partitions themselves, as mathematical objects, feature several competing coding conventions, and it is also partly due to the exchangeability properties often enjoyed by the many alternate representations of the DP.

In this chapter we discuss two encoding methods for DP cluster membership indicators, and how they are equivalent in terms of the partition that they identify, yet one holds more informative power than the other with respect to the information that it implicitly carries.

In particular, we compare cluster encoding according to the *order of appearance* of the clusters in the data against encoding the same in the order whereby their originating sticks are broken in the stick-breaking process. The latter encoding method, although ubiquitous, carries no specific name in the Bayesian nonparametric literature, and we hereby name it *stick-breaking order* encoding. Most importantly, we devise a way to translate cluster membership expressed in each coding convention into the other,

and to leverage it for the purpose of developing a new sampler for the Dirichlet process mixture model.

7.1 Encodings

In its most basic form of equation 2.1.1, a Dirichlet process is defined by the random probability masses assigned to its atoms, and by their random locations θ_i, which are sampled from G_0. Repeated values of θ_i identify random clusters of observations; often, inference on these clusters is one of the primary objectives of resorting to a DP or DPM in the first place. It is common practice to augment the model specification from equation 2.1.1 with additional latent variables, to express cluster membership information in isolation from the locations of the atoms – for example, to enable algorithms which are primarily based on the random partition induced by the DP, or to accompany applications where the location of the atoms is not of primary interest. There are multiple ways and conventions to encode cluster membership information into the latent cluster membership indicator vector.

7.1.1 Encoding in order of appearance

The original article pertaining to the Polya urn representation of the Dirichlet Process (Blackwell & MacQueen, 1973) ignores the topic of the location of the atoms of the DP, and only focuses on its probability masses and on the partition that the Polya urn sequence induces. The article only refers to the balls in the Polya urn having a certain *colour* x, and it does not point to any specific labelling convention for it. In this respect, x could belong to any set, as long as the set is countably infinite; in principle the set could even be a set of words or descriptors, to identify infinite colours, although doing so would be impractical, as no natural language that we know of has an infinite set of descriptors to match an infinite palette.

A more practical way, although certainly not the only way, to encode the information

about the colours of the balls extracted from the urn is to rely on natural numbers. This is for example how de Finetti et al., 2017 discusses the 2-colour Polya urn, where he assigns 1 to indicate a white ball, and 0 to indicate black. Similarly, colour membership in an n-colour Polya urn can be expressed with integers from $\mathbb{N}_n$, to map to specific colours. However, as outlined, the strategy of listing all possible colours and mapping them to $\mathbb{N}$ does not work well with the Dirichlet process due to its infinite dimensionality.

Common practice in the infinite-dimensional Polya urn setting is therefore to abandon the colour analogy entirely, and to encode cluster membership via $\mathbb{N}$, with the clusters labelled in the order whereby they appear in the sampling process (*order of appearance* – see section 2.3). Given an n-dimensional sample, cluster membership is encoded via vector $\boldsymbol{s} = (s_1, \ldots, s_n)$, defined so that $A_j \equiv \{i : s_i = j, i \in \mathbb{N}_n\}$. This results in a scheme where $s_1 = 1$, always; $s_2 = 1$ if the second ball from the urn has the same colour as the first ball, and $s_2 = 2$ otherwise, and so on:

$$
\begin{aligned}
&p\left(s_1 = 1\right) = 1, \\
&p\left(s_i \mid s_{i-1}, \ldots, s_1\right) = \frac{\alpha}{\alpha + i - 1}\delta_{k_{i-1}+1}\left(s_i\right) + \sum_{l=1}^{k_{i-1}} \frac{n_{i-1,l}}{\alpha + i - 1}\delta_l\left(s_i\right),
\end{aligned}
$$

where k_{i-1} and $n_{i-1,l}$ are respectively the number of clusters and the size of cluster l over the first $i-1$ observations.

7.1.2 Encoding in stick-breaking order

Just like the Polya urn representation, stick-breaking construction in the original article where it first appeared (Sethuraman, 1994) is not accompanied by indications of how to encode the sequence of its latent cluster membership indicators; in fact, the article relies on ties of the location of the point masses of the DP to implicitly identify its clusters.

In the DP literature, we mainly see two approaches to encoding the clustering scheme arising from the stick-breaking process. The first assigns integers to the

cluster membership indicator vector from $\mathbb{N}_k$ in the order of appearance of the sticks that the observations are sampled from. For example, conditional to $\boldsymbol{w}$, if the first drawn observation originates from the stick of length w_h, then it and any other subsequent draws from the same stick are assigned the label 1; the next observation to be drawn from any stick other than w_h, and any other subsequent observations drawn from the same, are assigned the label 2, and so on. The resulting sequence of cluster membership indicators is clearly equivalent to the one arising from a Polya urn, and it is labelled *in order of appearance* (the same discussed in section 7.1.1). For example, this type of encoding is used in the many influential articles from Pitman, where he discusses the stick-breaking process (for example, see Pitman, 2002, section 3.1).

The other approach involves indexing the sticks in their order of construction, and then assigning to each observation the index of the stick that the observation was sampled from. In symbols, consider equation 2.1.4, and express cluster membership through the vector $\boldsymbol{r} = (r_1, \ldots, r_n)$, where

$$p\left(r_i = h \mid \boldsymbol{w}\right) = w_h, \quad h = 1, 2, \ldots$$

While it is ubiquitous in the Bayesian nonparametric literature, this encoding method carries no name to identify it, and we hereby name it *stick-breaking order*.

We observe that encoding in stick-breaking order carries more information than encoding in order of appearance. While an ordered vector $\boldsymbol{r}$ encoded in stick-breaking order can always be deterministically brought back to its order of appearance ($\boldsymbol{s}$), by checking which elements of $\boldsymbol{r}$ are new as one progresses through the ordered sequence, the reverse is not true, and inference is needed to attempt to derive $\boldsymbol{r}$ back from $\boldsymbol{s}$.

7.1.3 Exchangeability

One of the main properties generally associated with the Polya urn construction of the Dirichlet process is that it produces "exchangeable sequences" and, while this is true under certain conditions, we would like to clarify the impact that different types of encoding have on this key property.

The finite Polya urn scheme, with a fixed number colours, is known to generate a sequence of random cluster membership indicators that is exchangeable (in the sense of de Finetti), and which is conditionally independent on a Dirichlet probability measure (Hill et al., 1987), when it is coded according to a finite set with elements corresponding to each colour. Similarly, due to the results summarised in section 2.1.1, the infinite-dimensional Polya urn is also known to produce sequences of random cluster membership indicators which are exchangeable, and which are conditionally independent on a Dirichlet process measure, as long as each of their labels uniquely identifies a specific colour. However, we observe that encoding a Polya urn sequence with the *order of appearance* encoding breaks its de Finetti exchangeability, due to the order constraints that the encoding imposes. For example, assume a Polya urn sequence with parameter α: the random vector $(s_1, s_2, \ldots)$ that it induces is not exchangeable as $s_1 = 1$ almost surely, while $p(s_2 = 1) < 1$, hence $p(s_1, s_2) \neq p(s_2, s_1)$. It can be seen that the conditional $\boldsymbol{s} \mid \boldsymbol{r}$ is not exchangeable either.

Conversely, stick-breaking encoding does lead to a sequence $\boldsymbol{r}$ that is de Finetti exchangeable, as r_i, r_j are conditionally independent on $\boldsymbol{w}$. However, its conditional $\boldsymbol{r} \mid \boldsymbol{s}$ is only de Finetti partially exchangeable – it is not fully exchangeable.

What is preserved under both encoding methods is the exchangeability of the random partition induced by the Dirichlet process. Both encoding methods lead to a random exchangeable partition.

We summarise these properties in table 7.1.

property	$\boldsymbol{s}$	$\boldsymbol{s} \mid \boldsymbol{r}$	$\boldsymbol{r}$	$\boldsymbol{r} \mid \boldsymbol{s}$
exchangeability	no	no	yes	no
partial exchangeability	no	no	yes	yes
exchangeability of its partition	yes	yes	yes	yes

Table 7.1: Exchangeability of the cluster membership indicator vector when encoded in order of appearance ($\boldsymbol{s}$) and in stick-breaking order ($\boldsymbol{r}$), and of its posterior.

7.2 A bridge between encodings

In a set of n observations with $K = k_n$ observed clusters, denote the distinct values of $\boldsymbol{r}$ (in the order whereby they appear in the vector $\boldsymbol{r}$) by $\left(r_1^\star, \ldots, r_{k_n}^\star\right)$, as follows:

$$
\begin{aligned}
r_1^\star &= r_1, \\
r_j^\star &= r_{\min\left\{i \in \mathbb{N}_n:\ r_i \notin \left\{r_1^\star, \ldots, r_{j-1}^\star\right\}\right\}}, \quad j = 2, \ldots, k_n.
\end{aligned}
$$

The random vector $\boldsymbol{r}^\star$ carries precisely the extra information that is needed over $\boldsymbol{s}$ to retrieve $\boldsymbol{r}$, as the pair $(\boldsymbol{r}^\star, \boldsymbol{s})$ is informationally equivalent to $\boldsymbol{r}$. In what follows, we aim to construct a sampler which explicitly returns both $\boldsymbol{r}^\star$ and $\boldsymbol{s}$, so that both encodings can be seen as originating from the same construction process. To proceed, we further denote by $\tilde{\boldsymbol{w}}$ the lengths of the sticks in the order whereby the original sticks are discovered in the data sampling process. We also use the symbols $\tilde{\boldsymbol{n}} = (\tilde{n}_1, \ldots, \tilde{n}_k)$ to emphasize that the cluster sizes we refer to are also indexed according to the order whereby the clusters are discovered in the data sampling process.

The random vectors $\boldsymbol{r}, \boldsymbol{r}^\star, \boldsymbol{s}, \boldsymbol{w}, \tilde{\boldsymbol{w}}, \tilde{\boldsymbol{n}}$ can all be thought of as originating from a 3-step process which first entails the stick-breaking process, to obtain $\boldsymbol{w}$, then it entails sampling $\tilde{\boldsymbol{w}} \mid \boldsymbol{w}$ to decide the order of appearance in the sample of each original stick, and then again it involves sampling $\tilde{n}_1, \ldots, \tilde{n}_k \mid \tilde{\boldsymbol{w}}$, to determine the size of each observed cluster. Doing so equates to drawing $\boldsymbol{r}, \boldsymbol{r}^\star, \boldsymbol{s}, \boldsymbol{w}, \tilde{\boldsymbol{w}}, \tilde{\boldsymbol{n}}$ as follows:

1. sample $\boldsymbol{w} = (w_1, w_2, \ldots)$ as per equation 2.1.4. Practically, one does not need to sample all of the infinite elements of $\boldsymbol{w}$, but only those which are necessary

to perform all three steps as hereby described;

2. sample $\tilde{w}_1, \tilde{w}_2, \ldots \mid \boldsymbol{w}$, and therefore implicitly sample $r_1^\star, r_2^\star, \ldots \mid \boldsymbol{w}$ too, from the size-biased sampling equation:

$$p\left(\tilde{w}_1 = w_{r_1^\star}, \tilde{w}_2 = w_{r_2^\star}, \ldots \mid \boldsymbol{w}\right) = p\left(r_1^\star, r_2^\star, \ldots \mid \boldsymbol{w}\right) = \prod_{j=1}^{\infty} w_{r_j^\star} \prod_{j=2}^{\infty} \frac{1}{1 - \sum_{l<j} w_{r_l^\star}}.$$

This can be accomplished by first drawing $r_1^\star \mid \boldsymbol{w}$, then iteratively drawing $r_j^\star \mid \boldsymbol{w}, r_1^\star, \ldots, r_{j-1}^\star$ for $j = 2, 3, \ldots$ from the infinitely-dimensional truncated[1] categorical distribution with probabilities in $\left\{w_h : h \in \mathbb{N} \setminus \left\{r_1^\star, \ldots, r_{j-1}^\star\right\}\right\}$;

3. sample $s_1, \ldots, s_n \mid \tilde{\boldsymbol{w}}$, hence implicitly $\tilde{n}_1, \ldots, \tilde{n}_k \mid \tilde{\boldsymbol{w}}$ too, by sequentially drawing from the following, for $i = 2, \ldots, n$:

$$p\left(s_i \mid s_1, \ldots, s_{i-1}, \tilde{\boldsymbol{w}}\right) = \left(\sum_{j=1}^{k_{i-1}} \tilde{w}_j \delta_j\left(s_i\right)\right) + \left(1 - \sum_{l \leq k_{i-1}} \tilde{w}_l\right) \delta_{k_{i-1}+1}\left(s_i\right), \quad (7.2.1)$$

where k_{i-1} is the number of clusters in $s_1, \ldots, s_{i-1}$.

We derived equation 7.2.1 from Pitman, 2002, equation 3.4:

$$p\left(\tilde{n}_1, \ldots, \tilde{n}_k \mid \tilde{\boldsymbol{w}}\right) = \left(\prod_{j=1}^{k} \tilde{w}_j^{\tilde{n}_j - 1}\right) \left(\prod_{j=1}^{k-1} 1 - \sum_{l \leq j} \tilde{w}_l\right). \quad (7.2.2)$$

Intuitively, given $s_1, \ldots, s_{i-1}$ and $\tilde{\boldsymbol{w}}$, the probability of s_i opening up a new cluster is $1 - \sum_{l \leq k_{i-1}} \tilde{w}_l$ (see the second factor in equation 7.2.2); conversely, the probability of s_i repeating some already observed cluster j is $\tilde{w}_j$ (see the first factor in the same equation).

Ultimately, because at the end of step 3 one has obtained $\boldsymbol{r}^\star, \boldsymbol{s}, \boldsymbol{w}, \tilde{\boldsymbol{w}}, \tilde{\boldsymbol{n}}$, one also has $\boldsymbol{r}$, which is fully identified by $\boldsymbol{r}^\star, \boldsymbol{s}$.

An important consideration for the above is that $(w_1, w_2, \ldots)$ has the same probability distribution as $(\tilde{w}_1, \tilde{w}_2, \ldots)$, as proved in Hoppe, 1987.

[1] *Truncated* because of the removal of the previously sampled categories.

7.3 Mapping r to s

Recall that repeated values in $\boldsymbol{r}$ induce a random partition $\{A_1, \ldots, A_k\}$ on $\mathbb{N}_n$, where we define $A_1, \ldots, A_k$ in order of appearance, such that $1 \in A_1$ and for each $2 \leq i \leq k$ the first element of $\mathbb{N}_n \setminus (A_1 \cup \ldots \cup A_{i-1})$ belongs to A_i. The vector $\boldsymbol{s}$ is then entirely determined by $\boldsymbol{r}$ through the mapping $g : \mathbb{N}^n \mapsto \mathbb{N}_k^n$ induced by the relationship:

$$s_i = j \iff i \in A_j.$$

7.4 Inferring r from s

In this section, we discuss how to infer $\boldsymbol{r}$ from $\boldsymbol{s}$. We start from the simplest approach possible, which involves the accept/reject algorithm; we measure its acceptance rate, discuss its limitations and we improve on it via re-ordering, obtaining a much improved acceptance rate. We then move on to discuss an augmentation of the target space, which allows sampling with almost sure acceptance.

7.4.1 Accept-reject method 1

To infer $\boldsymbol{r}$ from $\boldsymbol{s}$, we write

$$p(\boldsymbol{r} \mid \boldsymbol{s}) \propto p(\boldsymbol{s} \mid \boldsymbol{r})\, p(\boldsymbol{r}), \tag{7.4.1}$$

and we observe that:

- $p(\boldsymbol{r})$ is easy to sample from, by progressively sampling from $p(r_1), p(r_2 \mid r_1), \ldots$ according to:

  $$p(r_1 = h) = \mathbb{E}[w_h] = \frac{\alpha^{h-1}}{(\alpha+1)^h},$$

  and

  $$p(r_i = h \mid r_1, \ldots, r_{i-1}) = \mathbb{E}[w_h \mid r_1, \ldots, r_{i-1}]$$

$$= \mathbb{E}\left[v_h \mid r_1, \ldots, r_{i-1}\right] \prod_{l<h} \mathbb{E}\left[1 - v_l \mid r_1, \ldots, r_{i-1}\right], \quad i = 2, 3, \ldots,$$

where the conditional expectations can be obtained from equation 3.4.2;

- the first term on the right hand side of equation 7.4.1 acts as an indicator function:

$$p\left(\boldsymbol{s} \mid \boldsymbol{r}\right) = \mathbb{1}_{g(\boldsymbol{r})}\left(\boldsymbol{s}\right).$$

Therefore, we can draw a proposal $\hat{\boldsymbol{r}}$ from $p\left(\boldsymbol{r}\right)$, and accept it if $g\left(\hat{\boldsymbol{r}}\right) = \boldsymbol{s}$, or repeat the attempt if otherwise. The resulting acceptance rate equals equation 2.3.3 (Ewens' distribution), which returns the probability of one specific configuration of $\boldsymbol{s}$ where the cluster sizes are $n_1, \ldots, n_k$:

$$\alpha^k \frac{\Gamma\left(\alpha\right)}{\Gamma\left(\alpha + n\right)} \prod_{j=1}^{k} \Gamma\left(n_j\right).$$

However, the probability above approaches zero very quickly as n increases, to the point of quickly becoming unusable; for example, for a sequence with $n = 30, \alpha = 1$ with 3 clusters of size $(22, 7, 1)$, its theoretical acceptance rate as derived via equation 2.3.3 is approximately 1.4×10^{-10}, hence the algorithm is quite wasteful.

7.4.2 Accept-reject method 2

Consider a permutation σ of $\mathbb{N}$, and observe that:

- while $\boldsymbol{s}$ is not de Finetti exchangeable (because of its order constraint), we still have that $p\left(s_1, \ldots, s_n\right) = p\left(s_{\sigma_1}, \ldots, s_{\sigma_n}\right)$ as long as σ does not alter the order of appearance of the clusters;

- $p\left(r_1, \ldots, r_n\right) = p\left(r_{\sigma_1}, \ldots, r_{\sigma_n}\right)$, for any σ, because the random vector $\boldsymbol{r}$ is de Finetti exchangeable.

It is therefore legitimate to relax the acceptance criterion from section 7.4.1 to accept

all cases where the ordered[2] vector of the cluster sizes of $\hat{\boldsymbol{r}}$ matches the ordered cluster sizes of $\boldsymbol{s}$; doing so increases the acceptance rate to

$$\alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \frac{n!}{n_k(n_k+n_{k-1})\cdots(n_k+\ldots+n_1)},$$

as per equation 2.3.5.

We exemplify this with the aid of tables 7.2 and 7.3, which shows the set of all possible outcomes of $\boldsymbol{s}$ when $n=3$. When, for example, $\boldsymbol{s}=(1,1,2)$, the algorithm from section 7.4.1 aims to capture all outcomes of $\boldsymbol{r}$ whose configuration is compatible with $\boldsymbol{s}=(1,1,2)$; there is only one such case in Table 7.2 (see row 2). The algorithm from this section, instead, moves from the observation that row 3 from Table 7.2 is entirely equivalent to row 2 of the same, once both $\boldsymbol{s}$ and $\boldsymbol{r}$ transformed via $\sigma_1=1,\sigma_2=3,\sigma_3=2$. In this specific example, doing so doubles the acceptance ratio; in the example from section 7.4.1 ($n=30,\alpha=1$; 3 clusters of size $22,7,1$), it improves the acceptance rate from 1.4×10^{-10} to 0.00416. Table 7.3 summarises the configurations of $\boldsymbol{r}$ that can be accepted, once appropriately re-arranged.

Row	$\boldsymbol{s}$	$p(\boldsymbol{s})$	acceptable $\boldsymbol{r}$ pattern
1	$(1,1,1)$	0.3333	(a,a,a)
2	$(1,1,2)$	0.1666	(a,a,b)
3	$(1,2,1)$	0.1666	(a,b,a)
4	$(1,2,2)$	0.1666	(a,b,b)
5	$(1,2,3)$	0.1666	(a,b,c)

Table 7.2: Accept-reject algorithm 1, acceptable configurations of $\boldsymbol{r}$ for all possible outcomes of $\boldsymbol{s}$, in a Dirichlet process where $n=3,\alpha=1$.

7.4.3 Accept-reject method 3

The approach from section 7.4.2 can be pushed even further by observing that $p(s_1,\ldots,s_n)=p(\tau(s_{\sigma_1}),\ldots,\tau(s_{\sigma_n}))$, for any permutations σ,τ which do not alter the order of appearance of the clusters in $(\tau(s_{\sigma_1}),\ldots,\tau(s_{\sigma_n}))$, as $\boldsymbol{s}$ is invariant

[2]With $\boldsymbol{r}$ encoded in stick-breaking order, and with the sizes of the clusters ordered so that the size of the first cluster to appear is positioned first, the size of the second cluster to appear is positioned second, etc.

Row	$\boldsymbol{s}$	$p(\boldsymbol{s})$	acceptable $\boldsymbol{r}$ pattern
1	$(1,1,1)$	0.3333	(a,a,a)
2	$(1,1,2)$	0.1666	$(a,a,b),(a,b,a)$
3	$(1,2,1)$	0.1666	$(a,b,a),(a,a,b)$
4	$(1,2,2)$	0.1666	(a,b,b)
5	$(1,2,3)$	0.1666	(a,b,c)

Table 7.3: Accept-reject algorithm 2, acceptable configurations of $\boldsymbol{r}$ for all possible outcomes of $\boldsymbol{s}$, in a Dirichlet process where $n=3,\alpha=1$.

to permutations of its labels (subject to the aforementioned condition). This is exemplified in Table 7.4.

The algorithm works by generating a proposal $\hat{\boldsymbol{r}}$, sorting its cluster sizes in increasing (or decreasing) order and comparing them with the sorted cluster sizes of $\boldsymbol{s}$; if they match, $\hat{\boldsymbol{r}}$ is accepted and the positions of its elements are permuted to match the pattern of $\boldsymbol{s}$.

The acceptance rate of this algorithm is (see section 2.3.1):

$$\frac{n!}{\prod_{i=1}^{n} M_i! \left(i!\right)^{M_i}} \cdot \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \prod_{j=1}^{k} \Gamma(n_j). \tag{7.4.2}$$

This is a moderate improvement to the algorithm from section 7.4.2; for example, in the same case as above ($n=30,\alpha=1$; 3 clusters of size $22,7,1$), its acceptance rate is 0.0065, up from 0.00416. However, as is to be expected, even this algorithm becomes more wasteful as n increases, and in fact, for a sample size of $n=320$ (such as for example the thumbtack dataset used in Liu, 1996), the acceptance rate is 1 in about 70 million, for cluster sizes $(226,75,13,3,2,1)$, which motivates the need to devise a better algorithm.

7.4.4 Posterior augmentation method

We introduce the indicator variable $\boldsymbol{t}=(t_1,t_2,\ldots)$ to explicitly describe the order of appearance of $(w_1,w_2,\ldots)$:

$$t_j = i \Leftrightarrow r_i^{\star} = j. \tag{7.4.3}$$

Row	$\boldsymbol{s}$	$p(\boldsymbol{s})$	acceptable $\boldsymbol{r}$ pattern
1	$(1,1,1)$	0.3333	(a,a,a)
2	$(1,1,2)$	0.1666	$(a,a,b),(a,b,a),(a,b,b)$
3	$(1,2,1)$	0.1666	$(a,b,a),(a,a,b),(a,b,b)$
4	$(1,2,2)$	0.1666	$(a,b,b),(a,a,b),(a,b,a)$
5	$(1,2,3)$	0.1666	(a,b,c)

Table 7.4: Accept-reject algorithm 3, acceptable configurations of $\boldsymbol{r}$ for all possible outcomes of $\boldsymbol{s}$, in a Dirichlet process where $n = 3$.

Our motivating idea is that the pair $(\tilde{\boldsymbol{w}}, \boldsymbol{t})$ expresses the same information as $(\boldsymbol{w}, \boldsymbol{r}^\star)$; in doing so, we took inspiration from Lee et al., 2013, section 3.2, and we reversed it.

Now that $\boldsymbol{t}$ is defined, we can draw $\boldsymbol{r}^\star \mid \boldsymbol{s}$ by sampling from the joint posterior $p(\boldsymbol{r}^\star, \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}} \mid \boldsymbol{s})$:

$$\begin{aligned} p(\boldsymbol{r}^\star, \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}} \mid \boldsymbol{s}) &= p(\boldsymbol{r}^\star \mid \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}}, \boldsymbol{s}) \cdot p(\boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}} \mid \boldsymbol{s}) \\ &= p(\boldsymbol{r}^\star \mid \tilde{\boldsymbol{w}}, \boldsymbol{t}) \cdot p(\boldsymbol{w}, \boldsymbol{t} \mid \tilde{\boldsymbol{w}}, \boldsymbol{s}) \cdot p(\tilde{\boldsymbol{w}} \mid \boldsymbol{s}) \\ &= p(\boldsymbol{r}^\star \mid \tilde{\boldsymbol{w}}, \boldsymbol{t}) \cdot p(\boldsymbol{w}, \boldsymbol{t} \mid \tilde{\boldsymbol{w}}) \cdot p(\tilde{\boldsymbol{w}} \mid \boldsymbol{s}), \end{aligned} \tag{7.4.4}$$

where the first term in equation 7.4.4 is an indicator function which does not require any sampling, as $\boldsymbol{r}^\star \mid \tilde{\boldsymbol{w}}, \boldsymbol{t}$ is entirely determined by $(\tilde{\boldsymbol{w}}, \boldsymbol{t})$, while the second and third term need to be sampled.

The algorithm involves three steps (in the reverse order of the three factors from equation 7.4.4):

1. sample $\tilde{\boldsymbol{w}} \mid \boldsymbol{s}$,
2. sample $\boldsymbol{w}, \boldsymbol{t} \mid \tilde{\boldsymbol{w}}$,
3. deterministically obtain $p(\boldsymbol{r}^\star \mid \tilde{\boldsymbol{w}}, \boldsymbol{t})$ and therefore $\boldsymbol{r} \mid \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}}, \boldsymbol{s}$.

Although the algorithm does entail infinite-length vectors, from a practical perspective only a finite number of elements needs to be drawn, hence the algorithm can still be executed exactly.

Sampling $\tilde{\boldsymbol{w}} \mid \boldsymbol{s}$

The posterior distribution of $\tilde{\boldsymbol{w}} \mid \boldsymbol{s}$ can be expressed as

$$\begin{aligned} p\left(\tilde{\boldsymbol{w}} \mid \boldsymbol{s}\right) &= \frac{p\left(\boldsymbol{s} \mid \tilde{\boldsymbol{w}}\right) p\left(\tilde{\boldsymbol{w}}\right)}{p\left(\boldsymbol{s}\right)} \\ &= \frac{1}{p\left(\boldsymbol{s}\right)} \left(\prod_{i=1}^{k} \tilde{w}_i^{\tilde{n}_i - 1} \right) \left(\prod_{i=1}^{k-1} \left(1 - \sum_{j=1}^{i} \tilde{w}_j \right) \right) \frac{\alpha^k \left(1 - \tilde{w}_1 - \ldots - \tilde{w}_k\right)^{\alpha - 1}}{\left(1 - \tilde{w}_1\right) \ldots \left(1 - \tilde{w}_1 - \ldots - \tilde{w}_{k-1}\right)} \\ &= \frac{\alpha^k}{p\left(\boldsymbol{s}\right)} \left(\prod_{i=1}^{k} \tilde{w}_i^{\tilde{n}_i - 1} \right) \left(1 - \tilde{w}_1 - \ldots - \tilde{w}_k\right)^{\alpha - 1}, \end{aligned} \tag{7.4.5}$$

where the second passage above leverages corollary 7 from Pitman, 1995 and the fact that:

$$p\left(\tilde{n}_1, \ldots, \tilde{n}_k \mid \tilde{\boldsymbol{w}}\right) = \left(\prod_{i=1}^{k} \tilde{w}_i^{\tilde{n}_i - 1} \right) \left(\prod_{i=1}^{k-1} \left(1 - \sum_{j=1}^{i} \tilde{w}_j \right) \right). \tag{7.4.6}$$

We recall that $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$ are equal in distribution. As the former can be expressed in terms of $\boldsymbol{v}$, where each element is a priori beta-distributed, then the latter also can, and we name the equivalent terms $\tilde{\boldsymbol{v}}$.

Recalling that the determinant $|J|$ of the Jacobian of the transform

$$v_i = \frac{w_i}{1 - \sum_{l<i} w_l}$$

is

$$|J| = \prod_{i=1}^{k-1} \left(1 - v_i\right)^{k-i},$$

we apply it to derive $p\left(\tilde{\boldsymbol{v}} \mid \boldsymbol{s}\right)$ from $p\left(\tilde{\boldsymbol{w}} \mid \boldsymbol{s}\right)$:

$$\begin{aligned} p\left(\tilde{\boldsymbol{v}} \mid \boldsymbol{s}\right) &= \frac{\alpha^k}{p\left(\boldsymbol{s}\right)} \left(\prod_{i=1}^{k} \tilde{v}_i^{\tilde{n}_i - 1} \prod_{l<i} \left(1 - \tilde{v}_l\right)^{\tilde{n}_i - 1} \right) \left(\prod_{i=1}^{k} 1 - \tilde{v}_i \right)^{\alpha - 1} \prod_{i=1}^{k-1} \left(1 - \tilde{v}_i\right)^{k-i} \\ &= \frac{\alpha^k}{p\left(\boldsymbol{s}\right)} \left(\prod_{i=1}^{k} \tilde{v}_i^{\tilde{n}_i - 1} \prod_{l<i} \left(1 - \tilde{v}_l\right)^{\tilde{n}_i - 1} \right) \left(1 - \tilde{v}_k\right)^{\alpha - 1} \prod_{i=1}^{k-1} \left(1 - \tilde{v}_i\right)^{\alpha + k - i - 1} \\ &= \frac{\alpha^k}{p\left(\boldsymbol{s}\right)} \left(\prod_{i=1}^{k} \tilde{v}_i^{\tilde{n}_i - 1} \right) \prod_{i=1}^{k} \left(1 - \tilde{v}_i\right)^{\sum_{l>i} \tilde{n}_l + \alpha - 1}, \end{aligned}$$

which leads to:

$$\tilde{v}_i \mid \boldsymbol{s} \sim \text{Beta}\left(\tilde{n}_i, \alpha + \sum_{l>i} \tilde{n}_l \right).$$

This provides the joint distribution of $\tilde{\boldsymbol{w}} \mid \boldsymbol{s}$ and $\tilde{\boldsymbol{v}} \mid \boldsymbol{s}$. Their marginals are consistent

(as they should be) with the result from Hoppe, 1987, according to which $\tilde{w}_1 \mid \boldsymbol{s} \sim \text{Beta}\left(\tilde{n}_1, \alpha + n - \tilde{n}_1\right)$ and, by exchangeability,

$$\tilde{w}_i \mid \boldsymbol{s} \sim \text{Beta}\left(\tilde{n}_i, \alpha + n - \tilde{n}_i\right),$$

as can also be obtained in our setting from equation 7.4.5 by integrating out the nuisance terms. This also dovetails with Ferguson, 1973, theorem 1 (also explained in Pitman, 1996b, theorem 1, which better contextualises it with $\tilde{\boldsymbol{w}}$), according to which

$$\left(\tilde{w}_1, \ldots, \tilde{w}_k, 1 - \sum_{j=1}^{k} \tilde{w}_j\right) \sim \text{Dirichlet}\left(\tilde{n}_1, \ldots, \tilde{n}_k, \alpha\right), \tag{7.4.7}$$

by definition of a Dirichlet process.

Sampling $\boldsymbol{w}, \boldsymbol{t} \mid \tilde{\boldsymbol{w}}$

Because $\boldsymbol{w}$ is invariant under size-biased permutations (see Pitman, 1996a, and see also section 2.1.2 of this thesis), we have that $\boldsymbol{w} \mid \tilde{\boldsymbol{w}} \stackrel{d}{=} \tilde{\boldsymbol{w}} \mid \boldsymbol{w}$, and the process of sampling $\boldsymbol{w} \mid \tilde{\boldsymbol{w}}$ is the same as sampling $\tilde{\boldsymbol{w}} \mid \boldsymbol{w}$, with the labels replaced. Intuitively, $\boldsymbol{w} \mid \tilde{\boldsymbol{w}}$ and $\tilde{\boldsymbol{w}} \mid \boldsymbol{w}$ are both size-biased random permutations of the same unordered set of weights. Sampling $\tilde{\boldsymbol{w}} \mid \boldsymbol{w}$ is obtained via size-biasing through equation 2.1.8; the equation to sample from $\boldsymbol{t} \mid \tilde{\boldsymbol{w}}$ can be easily obtained from equation 2.1.8 by relabelling its arguments:

$$p\left(t_1 = \tau \mid \tilde{\boldsymbol{w}}\right) = \tilde{w}_\tau, \quad \tau = 1, 2, \ldots,$$
$$p\left(t_j = \tau \mid t_1, \ldots, t_{j-1}, \tilde{\boldsymbol{w}}\right) = \frac{\tilde{w}_\tau \; \mathbb{1}\left[\tau \neq t_i, \text{ for } 1 \leq i \leq j-1\right]}{1 - \tilde{w}_{t_1} - \ldots - \tilde{w}_{t_{j-1}}}, \quad j > 1, \tau = 1, 2, \ldots.$$

Obtaining $\boldsymbol{t} \mid \tilde{\boldsymbol{w}}$ also means obtaining $\boldsymbol{w} \mid \tilde{\boldsymbol{w}}$, as, by definition (see equation 7.4.3):

$$w_h = \tilde{w}_{t_h}.$$

Obtaining $\boldsymbol{r} \mid \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}}, \boldsymbol{s}$

As above, $\boldsymbol{r}^\star \mid \ldots$ can be retrieved from :

$$r_j^\star = \min \{i : t_i = j\}, \quad j = 1, 2, \ldots \tag{7.4.8}$$

Then, $\boldsymbol{r} \mid \ldots$ can be derived from $\boldsymbol{s}, \boldsymbol{r}^\star$.

Overall, the posterior augmentation approach is clearly superior to the accept-reject methods outlined in sections 7.4.1 to 7.4.3, as it delivers i.i.d. samples without ever being rejected. We name it the "transcoding algorithm".

7.5 Test

To showcase the capabilities of the posterior augmentation method, we carry out the following experiment:

- we sample a vector $\boldsymbol{r}$ of length 5 from the stick-breaking process with parameter α, 1 million times;
- we re-encode all sample paths of $\boldsymbol{r}$ into $\boldsymbol{s}$, as described in section 7.3, and we discard from $\boldsymbol{s}$ and $\boldsymbol{r}$ all sample paths where $\boldsymbol{s} \neq (1, 1, 1, 1, 2)$. By doing so, we are left with $50,090$ data points from a sample from the probability distribution $p(\boldsymbol{r} \mid \boldsymbol{s} = (1, 1, 1, 1, 2))$;
- we run the posterior augmentation algorithm on $\boldsymbol{s} = (1, 1, 1, 1, 2)$, to infer back $\boldsymbol{r} \mid \boldsymbol{s} = (1, 1, 1, 1, 2)$, and we compare with the frequencies obtained as above. Clearly, the two should match. We compare both marginal and joint frequencies.

Our results are in table 7.5. As expected, the probabilities match.

h	$p(r_1 = h \mid \boldsymbol{s} = (1,1,1,1,2))$ transcoder	empirical
1	0.6660	0.6670
2	0.2449	0.2432
3	0.0677	0.0682
4	0.0162	0.0164
5	0.0039	0.0040
6	0.0009	0.0010
7	0.0003	0.0001
8	0.0001	0.0000
...	...	...

h	$p(r_5 = h \mid \boldsymbol{s} = (1,1,1,1,2))$ transcoder	empirical
1	0.1659	0.1666
2	0.3592	0.3638
3	0.2281	0.2286
4	0.1219	0.1184
5	0.0635	0.0604
6	0.0304	0.0306
7	0.0156	0.0154
8	0.0080	0.0079
...	...	...

$\boldsymbol{h}$	$p(\boldsymbol{r} = \boldsymbol{h} \mid \boldsymbol{s} = (1,1,1,1,2))$ transcoder	empirical
$(1,1,1,1,2)$	0.3316	0.3350
$(1,1,1,1,3)$	0.1671	0.1679
$(2,2,2,2,1)$	0.1326	0.1336
$(1,1,1,1,4)$	0.0838	0.0823
$(2,2,2,2,3)$	0.0561	0.0560
$(1,1,1,1,5)$	0.0426	0.0401
$(2,2,2,2,4)$	0.0280	0.0265
$(3,3,3,3,1)$	0.0266	0.0268
...	...	...

Table 7.5: Marginal and joint posterior distribution of $\boldsymbol{r} \mid \boldsymbol{s} = (1,1,1,1,2)$, obtained via 1 million simulations from $\boldsymbol{r}$ ("empirical") and via $100,000$ iterations from the transcoder sampler ("transcoder"), for $\alpha = 1$.

7.6 The transcoding sampler

We move to the task of developing a sampler for the joint posterior of all of the parameters of interest in the stick-breaking construction of a Dirichlet process mixture: $\boldsymbol{r}$, $\boldsymbol{w}$ and $\boldsymbol{m}$. We write and factor the full joint posterior as:

$$\begin{aligned} p(\boldsymbol{r}, \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{s} \mid \boldsymbol{y}) &= p(\boldsymbol{s} \mid \boldsymbol{y}) \cdot p(\boldsymbol{r} \mid \boldsymbol{s}, \boldsymbol{y}) \cdot p(\boldsymbol{w} \mid \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{y}) \cdot p(\boldsymbol{m} \mid \boldsymbol{w}, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{y}) \\ &= p(\boldsymbol{s} \mid \boldsymbol{y}) \cdot p(\boldsymbol{r} \mid \boldsymbol{s}, \boldsymbol{y}) \cdot p(\boldsymbol{w} \mid \boldsymbol{r}, \boldsymbol{y}) \cdot p(\boldsymbol{m} \mid \boldsymbol{r}, \boldsymbol{y}) . \end{aligned}$$

The first factor can, in principle, be obtained with any sampler, including collapsed and sequential importance samplers. The third and fourth factors can be obtained as normal in the slice sampler framework. The second factor can be obtained with

the transcoding algorithm, as

$$p(\boldsymbol{r} \mid \boldsymbol{s}, \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{s})\, p(\boldsymbol{r} \mid \boldsymbol{s})}{p(\boldsymbol{y} \mid \boldsymbol{s})} = p(\boldsymbol{r} \mid \boldsymbol{s}),$$

which the transcoding algorithm is capable of producing.

Hence the transcoding algorithm can be used as a building block to form the *transcoding sampler*, where the full joint conditional is:

$$p(\boldsymbol{r}^\star, \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{t}, \tilde{\boldsymbol{w}}, \boldsymbol{s} \mid \boldsymbol{y}) = p(\boldsymbol{s} \mid \boldsymbol{y}) \cdot p(\boldsymbol{r}^\star, \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}} \mid \boldsymbol{s}) \cdot p(\boldsymbol{m} \mid \boldsymbol{r}^\star, \boldsymbol{s}, \boldsymbol{y}), \quad (7.6.1)$$

where the first factor can be produced with any sampler (Gibbs or SIS, for example), and the second factor can be produced with the transcoding algorithm. We henceforth refer to the sampling algorithm for $\boldsymbol{s} \mid \boldsymbol{y}$ as the *core sampler* of the transcoding sampler. Not only can we use as a core sampler one that produces $\boldsymbol{s} \mid \boldsymbol{y}$ directly, but we can also use any other sampler that produces $\boldsymbol{r} \mid \boldsymbol{y}$, which can always be mapped back to $\boldsymbol{s} \mid \boldsymbol{y}$ via the method outlined in section 7.3.

The third factor in equation 7.6.1 may not need sampling, as in often cases it is already produced by the core sampler in the first place. For example, collapsed algorithms 1 and 3 do generate the posterior of $\boldsymbol{\theta}$ alongside $\boldsymbol{s}$, and for occupied clusters we have

$$m_{r_i} = \theta_i, \quad i = 1, \ldots, n, \quad (7.6.2)$$

while for unoccupied clusters we have

$$m_j \sim G_0, \quad j \notin \{r_1, \ldots, r_n\}. \quad (7.6.3)$$

In case the core sampler does not return the posterior of $\boldsymbol{m}$, it can be obtained via the procedure outlined in section 6.4.1 and in equation 6.4.3 in particular.

To summarise, the transcoding sampler is composed of the following steps:

1. use the core sampler to generate posterior samples from $\boldsymbol{s} \mid \boldsymbol{y}$;

2. use the transcoding algorithm to sample from $\boldsymbol{r}, \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}} \mid \boldsymbol{s}$;

3. if the core sampler also provides the posterior of $\boldsymbol{\theta}$, obtain $\boldsymbol{m} \mid \boldsymbol{\theta}, \boldsymbol{r}, \ldots$ by re-ordering, via equations 7.6.2 and 7.6.3, or otherwise generate $\boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{r}$ via equation 6.4.3.

To stress the difference between $\boldsymbol{w} \mid \boldsymbol{y}$ and $\tilde{\boldsymbol{w}} \mid \boldsymbol{y}$, we include figure 7.1; while the a priori probability distribution of $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$ is the same, their posterior distributions are not.

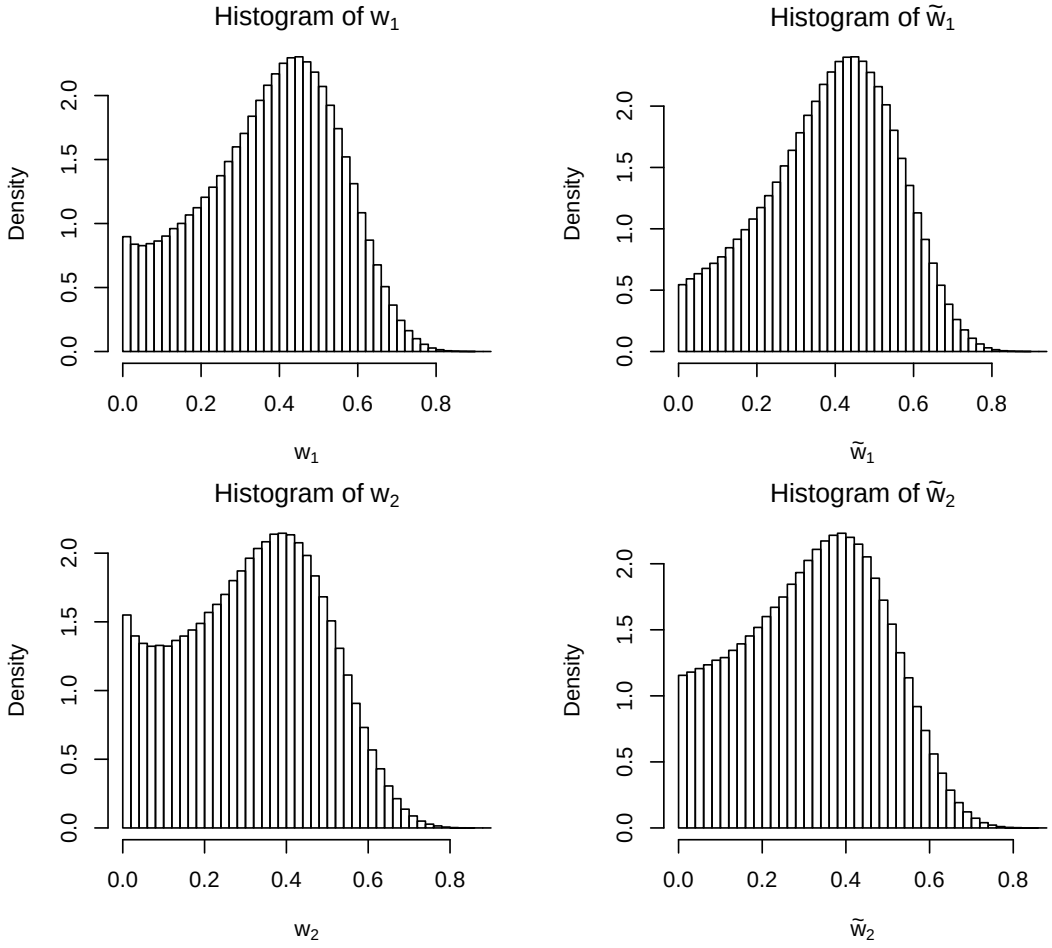

Figure 7.1: Histogram of the posterior of $w_1, \tilde{w}_1, w_2, \tilde{w}_2$, obtained with the collapsed algorithm 2 and the transcoder sampler, with $\alpha = 1$ and 2,000,000 iterations, on the thumb tack data set.

Other uses of the transcoder algorithm are certainly possible; for example, although it may not practically be useful, one could in principle create a Metropolis jump like the ones we describe in Chapter 5, by:

- running the slice sampler;

- obtaining $\boldsymbol{s} \mid \ldots$ from $\boldsymbol{r} \mid \ldots$ via what we describe in section 7.3;
- creating a new proposal for $\boldsymbol{r} \mid \ldots$ via the transcoder algorithm and attempting a jump.

Other possible uses go beyond pure sampling and may entail solutions to label-switching – as it can convert unidentifiable labels into identifiable ones, and possibly it could also be explored if the algorithm can be adapted to finite mixtures, in the same vein as in De Blasi and Gil-Leyva, 2021.

In our view, the most compelling compounding choices for the core sampler within the transcoding algorithm are:

- the sequential importance sampler SIS S2, as it is i.i.d. and, of all three sequential samplers from Chapter 6, it is the most efficient. The benefit that algorithm R would normally hold over S2 (i.e. to generate $\boldsymbol{r} \mid \ldots$ instead of $\boldsymbol{s} \mid \ldots$) is irrelevant in the context of the transcoding sampler, as the transcoding algorithm deals with that. The main disadvantage of using a sequential importance sampler as the core sampler is that its performance depends on the order whereby the observations appear in the sample, with no known optimality criterion to pre-sort the data;
- the most efficient collapsed sampler available for the task at hand (depending, for example, on whether the model uses a conjugate family); this would normally be collapsed algorithm 2, or Algorithm 8 from Neal, 2000. Collapsed samplers are known to have smaller integrated autocorrelation times than slice samplers, notwithstanding the many accelerating Metropolis jumps that have been proposed so far.

We conclude by pointing out that the transcoding sampler allows to level up the informational content of its core sampler, to include inference on the extra parameters which constitute the stick-breaking construction process, and which would otherwise

not be possible in the collapsed space, while preserving the efficiency of its core sampler.

7.7 Performance comparison

By definition, the integrated autocorrelation time (IAT) of the transcoding sampler with respect to the parameters that are shared with its core sampler is exactly the same between the two – no empirical performance test is needed to reach this conclusion.

However, the IAT of the additional parameters introduced by the transcoding sampler over those provided by the core sampler needs some testing, the outcomes of which we report in table 7.6. As expected, where the core sampler is i.i.d., the transcoder sampler returns IAT times at the minimum end of the spectrum (i.e. ≈ 0.50) – it produces fully i.i.d. samples. Where the core sampler is instead a collapsed algorithm, the IAT of the shared parameters is exactly the one from the core sampler, whereas any additional parameters do exhibit a small degree of autocorrelation, which is however extremely less pronounced than the one of the slice sampler, no matter which Metropolis label switching move the latter adopts.

For completeness, we also report in table 7.7 the values of $p(r_1 = h \mid \boldsymbol{y})$, to show that they match the same probabilities obtained with the slice sampler accelerated with move 4.

Algorithm	ESS	IAT_K	IAT_{w_1}	IAT_{r_1}	$\text{IAT}_{w_{r_1}}$	IAT_{m_1}	IAT_{θ_1}	IAT_D
SIS S2+transcoder	143,927	0.50	0.50	0.50	0.50	0.50	0.50	0.50
SIS R	132,154	0.50	0.50	0.50	0.50	0.50	0.50	0.50
collapsed2+transc.	N.A.	11.86	5.97	2.49	7.73	0.50	0.55	2.15

Table 7.6: Comparison of the transcoding sampler with sequential importance sampling algorithm S2 and collapsed algorithm 2 as its core samplers, and sequential importance sampling algorithm R, over $2,000,000$ iterations, on the thumb tack data set, with $\alpha = 1$.

	$p(r_1 = h \mid \boldsymbol{y})$	
h	S2-transcoded	slice4
1	0.3853	0.3837
2	0.3191	0.3201
3	0.1679	0.1676
4	0.0738	0.0747
5	0.0306	0.0313
6	0.0133	0.0126
7	0.0055	0.0055
8	0.0025	0.0024
9	0.0011	0.0011
10	0.0005	0.0005
...	...	...

Table 7.7: Posterior distribution of r_1, over $2,000,000$ iterations, on the thumb tack data set, with $\alpha = 1$, obtained with the SIS S2 algorithm (and the transcoding algorithm), and with the slice sampler and move 4).

7.8 Relationship with other work

Our idea for the transcoding algorithm (in its posterior augmentation form) was triggered by equation 18 from Lee et al., 2013. We thought that $p(\tilde{n}_1, \ldots, \tilde{n}_k)$ can be obtained algorithmically by first simulating $\boldsymbol{w}$, then drawing $\tilde{\boldsymbol{w}} \mid \boldsymbol{w}$ via size-biasing, and then finally by using corollary 7 from Pitman, 1995 (i.e. equation 7.4.6); inspired by it, we attempted to derive a way back from $\tilde{\boldsymbol{w}}$ to $\boldsymbol{w}$, which led to the transcoding algorithm.

Ultimately, the three main building blocks of the transcoding sampler are:

1. size-biasing;
2. equation 7.4.6;
3. the derivation of $\boldsymbol{w}, \boldsymbol{t} \mid \tilde{\boldsymbol{w}}$, and equation 7.4.8.

We have identified two other published approaches which revolve around building blocks 1 and 2, yet they reach different conclusions from ours as they do not use building block 3.

One is Fall and Barat, 2014, where an algorithm is developed which, in essence, samples from

$$p\left(s_i \mid \tilde{\boldsymbol{w}}, \boldsymbol{\theta}^\star \ldots\right) \propto \tilde{w}_{s_i} p\left(y_i \mid \theta^\star_{s_i}\right), \quad i = 1, \ldots, n, \tag{7.8.1}$$

then it samples $\tilde{\boldsymbol{w}} \mid \boldsymbol{s}, \ldots$ via equation 7.4.7, and then it samples their locations, in a Gibbs scheme, where cluster membership indicators are re-encoded at each iteration so that they are in order of appearance. The space that this algorithm operates in is somewhere in between the one of collapsed algorithm 2, and the one of the slice sampler: it is wider than the former, because it also includes $\tilde{\boldsymbol{w}}$, while it is narrower than the latter, as its cluster membership encoding is in order of appearance. Its IAT times published in Fall and Barat, 2014 reflect the same, positioning the algorithm between collapsed algorithm 2 and the slice sampler in terms of performance. However, the posterior that this algorithm produces does not include either $\boldsymbol{r}$ or $\boldsymbol{w}$, which instead the slice sampler returns – as such, it is not a replacement for the slice sampler.

The other is De Blasi and Gil-Leyva, 2021[3], which essentially[4] also operates as indicated earlier, with the technical difference that instead of sampling $s_i \mid \ldots$ from 7.8.1 and re-encoding the labels in order of appearance at each iteration, it adds constraints to the sampling formula so that the resulting cluster membership indicators generated by the sampler are always admissible to begin with, by construction, hence no need to re-encode them. This algorithm operates in the same space as Fall and Barat, 2014, and as above its posterior does not include $\boldsymbol{r}$ or $\boldsymbol{w}$ either. As such, we would expect it to perform like Fall and Barat, 2014 does. However, according to table 1 in De Blasi and Gil-Leyva, 2021, its IAT time for K_n appears to be shorter than the one of the "marginal sampler", which we can offer no explanation for.

To the best of our knowledge, our approach is novel and has not been attempted before; it allows to fully derive all of the stick-breaking parameters which are normally

[3] Also related to Gil–Leyva and Mena, 2021.

[4] The algorithm also has other features and uses, including for example its applicability to finite mixture models.

of interest, including $\boldsymbol{r} \mid \ldots$ and $\boldsymbol{w} \mid \ldots$, contrary to the other approaches mentioned above.

7.9 Conclusions

In this chapter, we have discussed cluster membership encoding according to the order of appearance of the clusters in the data (as it happens in $\boldsymbol{s}$), and according to the order of their originating sticks in the stick-breaking construction process (as it happens in $\boldsymbol{r}$); we have determined that the latter carries more information than the former. We have worked out a simple, deterministic approach to derive $\boldsymbol{s} \mid \boldsymbol{r}$, and four increasingly efficient ways of inferring $\boldsymbol{r} \mid \boldsymbol{s}$ – the most efficient of which we named the *transcoding algorithm*.

We have derived the *transcoding sampler*, which fully integrates with any other sampler capable of returning the exchangeable posterior partition of the data (i.e. its *core sampler*), to infer back all of the stick-breaking parameters including $\boldsymbol{r}, \boldsymbol{w}, \tilde{\boldsymbol{w}}$ and others. Since the transcoding sampler leverages the transcoding algorithm, which is i.i.d., it inherits the integrated autocorrelation times of the core sampler that it relies on. The transcoding sampler thus makes it possible to make full posterior inferences on stick-breaking parameters while attaining minimal autocorrelation times (when using the sequential importance sampler at its core), or while attaining the same autocorrelation times as those from collapsed samplers (when using collapsed samplers at its core), which are known to be much shorter than those attained by the slice sampler and other stick-breaking samplers (even in the presence of Metropolis label-switching moves, to accelerate the slice sampler).

Chapter 8

Conclusions

The entirety of this thesis revolves around the stick-breaking construction of the Dirichlet process. To decide the distributional specification and parameters of the prior of its precision parameter, we turned away from the current practice of looking at the distribution that it induces on the prior of the random number of observed clusters, and we instead leveraged the relationship between the length of the first two sticks in its the stick-breaking representation. In doing so, we opened up a new way of thinking about the precision parameter α, which overcomes the limitations of the pre-existing methods from literature that we have highlighted, especially for large n.

Similarly, in the context of Monte Carlo inferential sampling, we devised new methods to sample the parameters which specifically pertain to stick-breaking, in a way that is more efficient than ever. The performance gains are material: the problem was to overcome the large integrated autocorrelation times that usually accompany stick-breaking samplers, and we have devised one method that is i.i.d. (SIS R), and another which allows to re-purpose any other better performing non-stick-breaking sampler so that it generates stick-breaking parameters too, while retaining its better performance (*transcoding sampler*). The main building block of the latter, the *transcoding algorithm*, also has the potential for other applications.

We have also corrected and improved some existing Metropolis acceleration jumps,

and proposed a new one, although in our view their use is now put into question by the emergence of the transcoding sampler, which dominates them. Ultimately, a comparison amongst all samplers is displayed in table 8.1.

Algorithm	ESS	IAT_K	IAT_{w_1}	IAT_{r_1}	$\text{IAT}_{w_{r_1}}$	IAT_{m_1}	IAT_{θ_1}	IAT_D
SIS S2+transcoder	143,927	0.50	0.50	0.50	0.50	0.50	0.50	0.50
SIS R	132,154	0.50	0.50	0.50	0.50	0.50	0.50	0.50
collapsed2+transc.	N.A.	11.86	5.97	2.49	7.73	0.50	0.55	2.15
slice, no moves	N.A.	75.16	126.00	43.70	36.00	388.12	0.87	6.43
slice, move 1	N.A.	72.16	121.15	32.15	34.34	38.69	0.87	6.65
slice, move 2	N.A.	65.39	73.59	32.96	32.44	226.31	0.86	6.39
slice, move 3	N.A.	57.37	34.38	13.64	29.63	6.15	0.86	6.24
slice, move 4	N.A.	59.96	35.02	13.68	29.08	6.33	0.85	6.29

Table 8.1: Integrated autocorrelation times obtained with various samplers, over $2,000,000$ iterations, on the thumb tack data set, with $\alpha = 1$.

8.1 Future research

Many of the results included in this thesis deal with the stick-breaking representation of the Dirichlet process, which has the advantage, over the Polya urn representation, of being easy to modify to cover a wider family of models – for example, by taking sticks distributed as a Beta (a, b) rather than as a Beta $(1, \alpha)$. Several generalisations of the Dirichlet stick-breaking process exist, and the most immediate direction for future research would be to assess to what extent and how the results obtained herein can be extended to wider families – for example, to the Pitman-Yor process and others. This applies both to the samplers we have developed (SIS R and the transcoding sampler), and to our decisioning method for the distributional choice and parameters of $p(\alpha \mid \boldsymbol{\eta})$ too.

Secondarily, our results pertaining to sequential importance sampling indicate that there is room for more work to be carried out, to determine how sequential importance sampling is influenced by the order of the data, and what the optimal permutation of the data is. Another possible direction for further study is that of more powerful Sequential Monte Carlo methods.

Finally, the transcoding algorithm holds a potentially wide range of applications, due to the fact that it can be used in combination with other methodologies, to complement them.

Bibliography

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, *2*(6), 1152–1174.

Arratia, R., Barbour, A. D. & Tavaré, S. (2000). The number of components in a logarithmic combinatorial structure. *The Annals of Applied Probability*, *10*(2), 331–361.

Arratia, R., Barbour, A. D. & Tavaré, S. (2003). *Logarithmic Combinatorial Structures: a Probabilistic Approach*. European Mathematical Society.

Beckett, L. & Diaconis, P. (1994). Spectral Analysis for Discrete Longitudinal Data. *Advances in Mathematics*, *103*(1), 107–128.

Blackwell, D. & MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, *1*(2), 353–355.

Blei, D. M. & Jordan, M. I. (2006). Variational Inference for Dirichlet Process Mixtures. *Bayesian Analysis*, *1*(1), 121–143.

Bush, C. A. & MacEachern, S. N. (1996). A semiparametric Bayesian model for randomised block designs. *Biometrika*, *83*(2), 275–285.

Connor, R. J. & Mosimann, J. E. (1969). Concepts of independence for proportions with a generalization of the Dirichlet distribution. *Journal of the American Statistical Association*, *64*(325), 194–206.

Crane, H. (2016). The Ubiquitous Ewens Sampling Formula. *Statistical Science*, *31*(1), 1–19.

Dahl, D. B. & Newcomb, S. (2022). Sequentially allocated merge-split samplers for conjugate Bayesian nonparametric models. *Journal of Statistical Computation and Simulation*, *92*(7), 1487–1511.

De Blasi, P. & Gil-Leyva, M. F. (2021). Gibbs sampling for mixtures in order of appearance: the ordered allocation sampler [preprint v2]. https://arxiv.org/abs/2107.08380

de Finetti, B. (2011). *Induzione e statistica: Lectures given at a Summer School of the Centro Internazionale Matematico Estivo (C.I.M.E.) held in Varenna (Como), Italy, June 1-10, 1959*. Springer.

de Finetti, B., Machi, A. & Smith, A. F. M. (2017). *Theory of probability: a critical introductory treatment*. John Wiley & Sons.

Dorazio, R. M. (2009). On selecting a prior for the precision parameter of Dirichlet process mixture models. *Journal of Statistical Planning and Inference*, *139*(9), 3384–3390.

Engen, S. (1975). A Note on the Geometric Series as a Species Frequency Model. *Biometrika*, *62*(3), 697–699.

Escobar, M. D. (1988). *Estimating the Means of Several Normal Populations by Nonparametric Estimation of the Distribution of the Means* (Doctoral dissertation). Yale University.

Escobar, M. D. (1994). Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, *89*(425), 268–277.

Escobar, M. D. & West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, *90*(430), 577–588.

Escobar, M. D. & West, M. (1998). Computing nonparametric hierarchical models. *Practical Nonparametric and Semiparametric Bayesian Statistics* (pp. 1–22). Springer.

Fall, M. D. & Barat, É. *Gibbs sampling methods for Pitman-Yor mixture models*. 2014. https://hal.archives-ouvertes.fr/hal-00740770v2

Fearnhead, P. (2004). Particle filters for mixture models with an unknown number of components. *Statistics and Computing*, *14*(1), 11–21.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, *1*(2), 209–230.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. *Recent Advances in Statistics: Papers in Honor of Herman Chernoff on his Sixtieth Birthday* (pp. 287–302). Academic Press.

Gelfand, A. E. & Smith, A. (1990). Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, *85*(410), 398–409.

Geman, S. & Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 721–741.

Ghosal, S. & Van der Vaart, A. (2017). *Fundamentals of Nonparametric Bayesian Inference*. Cambridge University Press.

Gil–Leyva, M. F. & Mena, R. H. (2021). Stick-Breaking Processes With Exchangeable Length Variables. *Journal of the American Statistical Association*.
https://doi.org/10.1080/01621459.2021.1941054

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, *82*(4), 711–732.

Green, P. J. & Richardson, S. (2001). Modelling Heterogeneity With and Without the Dirichlet Process. *Scandinavian Journal of Statistics*, *28*(2), 355–375.

Hastie, D. I. & Green, P. J. (2012). Model choice using reversible jump Markov chain Monte Carlo. *Statistica Neerlandica*, *66*(3), 309–338.

Hastie, D. I., Liverani, S. & Richardson, S. (2015). Sampling from Dirichlet process mixture models with unknown concentration parameter: mixing issues in large data implementations [Software package at

https://www.jstatsoft.org/article/view/v064i07]. *Statistics and Computing*, *25*(5), 1023–1037.

Hill, B. M., Lane, D. & Sudderth, W. (1987). Exchangeable urn processes. *The Annals of Probability*, *15*(4), 1586–1592.

Hjort, N. L., Holmes, C., Müller, P. & Walker, S. G. (2010). *Bayesian Nonparametrics* (Vol. 28). Cambridge University Press.

Hoppe, F. M. (1984). Pólya-like urns and the Ewens' sampling formula. *Journal of Mathematical Biology*, *20*(1), 91–94.

Hoppe, F. M. (1987). The sampling theory of neutral alleles and an urn model in population genetics. *Journal of Mathematical Biology*, *25*(2), 123–159.

Ishwaran, H. & James, L. F. (2001). Gibbs Sampling Methods for Stick-Breaking Priors. *Journal of the American Statistical Association*, *96*(453), 161–173.

Ishwaran, H. & Zarepour, M. (2000). Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika*, *87*(2), 371–390.

Jain, S. & Neal, R. (2004). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, *13*(1), 158–182.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, *186*(1007), 453–461.

Kalli, M., Griffin, J. E. & Walker, S. G. (2011). Slice sampling mixture models. *Statistics and Computing*, *21*(1), 93–105.

Kingman, J. F. C. (1975). Random Discrete Distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, *37*(1), 1–22.

Lee, J., Quintana, F. A., Müller, P. & Trippa, L. (2013). Defining Predictive Probability Functions for Species Sampling Models. *Statistical Science*, *28*(2).

Liu, J. S. (1994). The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, *89*(427), 958–966.

Liu, J. S. (1996). Nonparametric hierarchical Bayes via sequential imputations. *The Annals of Statistics*, *24*(3), 911–930.

Liu, J. S., Wong, W. H. & Kong, A. (1994). Covariance Structure of the Gibbs Sampler with Applications to the Comparisons of Estimators and Augmentation Schemes. *Biometrika*, *81*(1), 27.

Lo, A. Y. (1984). On a class of Bayesian nonparametric estimates: I. Density estimates. *The Annals of Statistics*, *12*(1), 351–357.

Lunn, D., Jackson, C., Best, N., Thomas, A. & Spiegelhalter, D. (2012). *The BUGS Book: A Practical Introduction to Bayesian Analysis*. CRC Press.

MacEachern, S. & Müller, P. (1998). Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, *7*(2), 223–238.

MacEachern, S. & Müller, P. (2012). Efficient MCMC schemes for robust model extensions using encompassing Dirichlet Process Mixture Models. In D. Insua & F. Ruggeri (Eds.), *Robust Bayesian Analysis*. Springer New York.

MacEachern, S. N. (1994). Estimating normal means with a conjugate style Dirichlet process prior. *Communications in Statistics – Simulation and Computation*, *23*(3), 727–741.

MacEachern, S. N., Clyde, M. & Liu, J. S. (1999). Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics*, *27*(2), 251–267.

McCloskey, J. W. (1965). *A Model for the Distribution of Individuals by Species in an Environment* (Doctoral dissertation). Michigan State University.

Miller, J. W. (2019). An elementary derivation of the Chinese restaurant process from Sethuraman's stick-breaking process. *Statistics & Probability Letters*, *146*, 112–117.

Müller, P. & Rodriguez, A. (2013). *Nonparametric Bayesian Inference*. Institute of Mathematical Statistics.

Müller, P., Quintana, F. A., Jara, A. & Hanson, T. (2015). *Bayesian Nonparametric Data Analysis*. Springer International Publishing.

Murugiah, S. & Sweeting, T. (2012). Selecting the precision parameter prior in Dirichlet process mixture models. *Journal of Statistical Planning and Inference*, *142*(7), 1947–1959.

Navarro, D. J., Griffiths, T. L., Steyvers, M. & Lee, M. D. (2006). Modeling individual differences using Dirichlet processes. *Journal of Mathematical Psychology*, *50*(2), 101–122.

Neal, R. (2000). Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, *9*(2), 249–265.

Papaspiliopoulos, O. (2008). A note on posterior sampling from Dirichlet mixture models.
http://wrap.warwick.ac.uk/35493/1/WRAP_papaspliiopoulos_08-20wv2.pdf

Papaspiliopoulos, O. & Roberts, G. O. (2008). Retrospective Markov Chain Monte Carlo Methods for Dirichlet Process Hierarchical Models [Software package at http://www.econ.upf.edu/~omiros/fast_conditional_mcmc_local.for]. *Biometrika*, *95*(1), 169–186.

Phadia, E. G. (2015). *Prior Processes and Their Applications*. Springer.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, *102*(2), 145–158.

Pitman, J. (1996a). Random discrete distributions invariant under size-biased permutation. *Advances in Applied Probability*, *28*(2), 525–539.

Pitman, J. (1996b). Some Developments of the Blackwell-MacQueen Urn Scheme. *Lecture Notes-Monograph Series*, *30*, 245–267.

Pitman, J. (2002). *Combinatorial Stochastic Processes*. Springer.

Quintana, F. A. (1998). Nonparametric Bayesian Analysis for Assessing Homogeneity in $k \times l$ Contingency Tables With Fixed Right Margin Totals. *Journal of the American Statistical Association*, *93*(443), 1140–1149.

R Core Team. (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Robert, C. P. & Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 639–650.

Sokal, A. (1997). Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms [Series Title: NATO ASI Series]. In C. DeWitt-Morette, P. Cartier & A. Folacci (Eds.), *Functional Integration* (pp. 131–192). Springer US.

Tierney, L. (1994). Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics*, *22*(4), 1701–1728.

Ulker, Y., Gunsel, B. & Cemgil, A. T. (2011). Annealed SMC Samplers for Nonparametric Bayesian Mixture Models. *IEEE Signal Processing Letters*, *18*(1), 3–6.

van Dyk, D. A. & Park, T. (2008). Partially Collapsed Gibbs Samplers: Theory and Methods. *Journal of the American Statistical Association*, *103*(482), 790–796.

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

Watterson, G. A. (1974). The sampling theory of selectively neutral alleles. *Advances in Applied Probability*, *6*(3), 463–488.

Watterson, G. A. (1976). The stationary distribution of the infinitely-many neutral alleles diffusion model. *Journal of Applied Probability*, *13*(4), 639–651.

West, M. & Escobar, M. D. (1993). *Hierarchical Priors and Mixture Models, with Application in Regression and Density Estimation*. Institute of Statistics; Decision Sciences, Duke University.

Yau, C. & Holmes, C. (2011). Hierarchical Bayesian nonparametric mixture models for clustering with variable relevance determination [Software package at https://sites.google.com/site/mixlasso/]. *Bayesian Analysis*, *6*(2), 329–351.