

Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models

Omiros Papaspiliopoulos*and Gareth O. Roberts†

Abstract

An alternative to the Escobar and West (1995) computational approach was recently introduced by Ishwaran and Zarepour (2000) for making inference for Dirichlet process hierarchical models: instead of integrating out the Dirichlet process to impute it and update it as part of the Gibbs sampling algorithm. However, since the Dirichlet process is infinite dimensional, the authors suggest finite approximations in order to carry out the computations. We show that such approximations are not necessary and we design an MCMC algorithm which samples from the exact posterior distribution of all quantities of interest. The success of the algorithm is based on the technique of retrospective sampling, which we introduce in this paper, and which can be found useful in other problems where simulation and inference for infinite dimensional processes is considered.

1 Introduction

The use of Dirichlet mixture models (DMMs for short) has become increasingly popular for use in semi-parametric inference. Application areas include density estimation (e.g Escobar and West, 1995; Müller et al., 1996), survival analysis (e.g Doss, 1994; Gelfand and Kottas, 2003), semi-parametric analysis of variance (e.g Bush and MacEachern, 1996), cluster analysis and partition modelling (e.g MacEachern and Müller, 1998; Petrone and Raftery, 1997; Quintana and Iglesias, 2003). This is largely due to the tractability afforded these models by innovative MCMC techniques which have been developed following the seminal work by Escobar and West (1995). A full description of available techniques will be given later in the paper.

Broadly speaking, there are two possible computational approaches which correspond to two different data augmentation schemes (see Section 3 for details). The marginal approach ‘integrates out’ the Dirichlet process and uses convenient Pólya urn representations within a Gibbs sampler to obtain posterior

*Medical Statistics Unit, Lancaster University, Lancaster LA1 4YF, UK

†Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YF, UK

samples. This is the approach introduced in Escobar (1994) and Escobar and West (1995), and has been substantially improved in MacEachern (1994), Bush and MacEachern (1996), Müller et al. (1996) and particularly in MacEachern and Müller (1998) and Neal (2000). Although simple to implement, the main drawback of this method is that a single-component updating Gibbs sampler is used to sample from a multivariate distribution of dependent variables regardless of the size of the dataset. Such sampler can have serious problems in moving large clusters around the posterior space, and it is therefore not always possible to use the method effectively on moderate sized datasets. The alternative conditional approach essentially augments the Dirichlet process and updates it within a Gibbs sampler. By doing so, the updated variables are partitioned in a small number (3 or 4) of groups, where the variables within each group are conditionally independent given the variables in the other groups. Thus, the Gibbs sampler can be very efficiently implemented. This approach has been advocated by Ishwaran and Zarepour (2000) and Ishwaran and James (2001), who found that it can lead to considerably more robust convergence properties than the existing algorithms. However, since the Dirichlet process cannot be finitely represented, Ishwaran and Zarepour (2000) suggest to approximate it using some kind of truncation of the Dirichlet measure for the practical implementation of this method. Although it is possible to control the error produced by such truncations (Ishwaran and Zarepour, 2000, 2002) it would be desirable to avoid approximations altogether.

This paper demonstrates how to construct an MCMC algorithm which has all the computational advantages of the conditional approach, without needing to resort to approximation arguments at all. We do this by using a technique we call *retrospective sampling*. The seeds of this idea can be found in (Roberts et al., 2003) where retrospective simulation was used to implement an MCMC algorithm which involved updating point processes with infinite number of points, although the concept is formally introduced in the current paper. Ongoing work (which we will report at a later date) will show that a far more involved probabilistic construction using retrospective sampling can even allow pure Gibbs sampling to be implemented for this problem. We believe that the techniques and methods developed in this paper can have impact on other problems involving the imputation of infinite dimensional latent processes. Since the writing of this paper, retrospective sampling has been used for exact simulation of diffusion processes, see Beskos and Roberts (2004).

Apart from being able to implement the conditional approach without approximation error, our approach clarifies which are the differences and the similarities between the conditional method of Ishwaran and Zarepour (2000) and the marginal approach of MacEachern and Müller (1998).

The paper is organised as follows. In Section 2, we introduce the Dirichlet mixture model, and also give a brief description of retrospective sampling as a way of sampling from Dirichlet process. Although simple, the example serves to introduce the ideas behind the more complex algorithms described later in the paper. Section 3 reviews existing MCMC marginal methods for fitting DMMs, and Section 4 introduces the conditional method. Section 5 derives the

posterior distributions of the parameters of interest and Section 6 introduces the sampler proposed in this paper which obtains samples from the posterior distribution of quantities of interest using retrospective sampling; Section 6.1 summarises a simulation study which compares our method with competitive existing approaches. Specifically, we test our algorithm to various simulated datasets of size 100 and 1000. Section 7 discusses the issue of updating the hyperparameters of the DMM, and the challenges of this task associated with the conditional method. We propose non-centred parametrisations and demonstrate results based on simulated datasets. Finally, Section 8 finishes with a discussion.

2 Dirichlet mixture models: basics and notation

This paper considers inference for models with the following (semi-parametric) hierarchical structure

$$\begin{aligned} Y_i | (X_i, \phi) &\stackrel{iid}{\sim} \pi(Y_i | X_i, \phi), \quad i = 1, \dots, n \\ X_i | P &\sim P \\ P | (\alpha, \Theta) &\stackrel{iid}{\sim} DP(\alpha H_\Theta). \end{aligned} \tag{1}$$

In this hierarchical setting, the observed data Y_1, \dots, Y_n are mutually independent conditionally on the unobserved random elements X_1, \dots, X_n . The distribution of Y_i belongs in some parametric family with parameters X_i and ϕ . The X_i s take values on a measurable space $(\mathcal{X}, \mathcal{B})$ and are treated as exchangeable, thus allowing for pooling of information among the data. The prior chosen in (1) is the Dirichlet process, denoted by $DP(\alpha H_\Theta)$, with concentration parameter α , which is a positive real number, and base distribution H_Θ , which is a probability measure on $(\mathcal{X}, \mathcal{B})$ and is characterised by some parameters Θ . A fully Bayesian approach requires prior elicitation to all hyperparameters, thus (1) is completed by specifying $\pi(\alpha, \Theta, \phi)$. A simple and widely explored special case of (1) is the Dirichlet mixture of Gaussians (Ferguson, 1983; Escobar, 1994; Escobar and West, 1995), which for real-valued data writes as

$$\begin{aligned} Y_i | (X_i, \phi) &\stackrel{iid}{\sim} N(X_i, \phi) \\ X_i | P &\stackrel{iid}{\sim} P \\ P | (\Theta, \alpha) &\sim DP(\alpha H_\Theta) \\ H_\Theta &\equiv N(\mu, \sigma_z^2) \\ \Theta &= (\mu, \sigma_z^2). \end{aligned} \tag{2}$$

(The notation used for the prior variance, σ_z^2 , will become obvious later.) One of the main strengths of the DMMs is their flexibility to be defined for arbitrary measure spaces $(\mathcal{X}, \mathcal{B})$. For example, a very interesting extension of (2) takes X_i to be two-dimensional, $X_i = (\mu_i, \sigma_i^2)$:

$$Y_i | X_i \stackrel{iid}{\sim} N(\mu_i, \sigma_i^2)$$

$$\begin{aligned}
X_i | P &\stackrel{iid}{\sim} P \\
P | (\Theta, \alpha) &\sim DP(\alpha H_\Theta) \\
H_\Theta &\equiv N(\mu, \sigma_z^2) \times \text{Ig}(\gamma, \beta) \\
\Theta &= (\mu, \sigma_z^2, \gamma, \beta),
\end{aligned} \tag{3}$$

where Ig denotes the inverse gamma distribution; see for example Müller et al. (1996); Green and Richardson (2001). Notice that when the DMM is used for density estimation the second model allows for local smoothing.

The Dirichlet process is a prior distribution on the space of all probability measures on $(\mathcal{X}, \mathcal{B})$, in the sense that each realisation of the process generates a measure on $(\mathcal{X}, \mathcal{B})$. Its name originates from its definition as a stochastic process (indexed by the elements B of \mathcal{B}), such that for every measurable partition (B_1, \dots, B_k) of \mathcal{X} ,

$$(P(B_1), \dots, P(B_k)) \sim \text{Dirichlet}\{\alpha H_\Theta(B_1), \dots, \alpha H_\Theta(B_k)\}$$

see Ferguson (1973), Walker et al. (1999) and Ishwaran and Zarepour (2002). From this definition it is easy to show that for every $A \in \mathcal{B}$,

$$E\{P(A)\} = H_\Theta(A), \quad \text{var}\{P(A)\} = \frac{H_\Theta(A)\{1 - H_\Theta(A)\}}{(1 + \alpha)}. \tag{4}$$

Nevertheless, a more constructive characterisation of the Dirichlet process is the *almost sure* infinite series representation given by Sethuraman (1994):

$$P = \sum_{j=1}^{\infty} p_j \delta_{Z_j}(\cdot), \tag{5}$$

where $\delta_x(\cdot)$ denotes the Dirac delta measure centred at x . The Z_j s are independent random variables with common distribution H_Θ , and the p_j s are defined iteratively as

$$p_1 = V_1, \quad p_j = (1 - V_1)(1 - V_2) \cdots (1 - V_{j-1})V_j, \quad j \geq 2 \tag{6}$$

or equivalently as

$$p_1 = V_1, \quad p_j = (1 - \Pi_{j-1})V_j, \quad j \geq 2, \tag{7}$$

where the V_j s are independent $\text{Be}(1, \alpha)$ random variables, and Π_j is the cumulative probabilities

$$\Pi_j = \sum_{k=1}^j p_k. \tag{8}$$

It is easy to show that the p_j s decay to zero almost surely at an exponential rate.

Representation (5) shows that every realisation of the Dirichlet process is a discrete measure on $(\mathcal{X}, \mathcal{B})$ with point masses $p_j \in (0, 1)$ at locations $Z_j \in$

\mathcal{X} . This prior assumption, that P is *almost surely* discrete, has some serious and unexpected effect on inferences, see for example Green and Richardson (2001) and Petrone and Raftery (1997). In the sequel, we will use the following notation: $p = (p_1, p_2, \dots)$, $Z = (Z_1, Z_2, \dots)$ and $V = (V_1, V_2, \dots)$.

The marginal dependence of the X_i s (i.e with P integrated out) is described by a well-known (Blackwell and MacQueen, 1973; Ferguson, 1973) Pólya urn scheme,

$$\begin{aligned} X_1 &\sim H_\Theta \\ X_n | X_1, \dots, X_{n-1} &\begin{cases} = X_j, & \text{with probability } 1/(\alpha + n - 1) \\ \sim H_\Theta, & \text{with probability } \alpha/(\alpha + n - 1). \end{cases} \end{aligned} \tag{9}$$

Therefore, conditionally on the previously sampled values, a new draw with some probability is exactly identical to one of the previous values and with the remaining probability is sampled independently from H_Θ . The prediction rule described by (2.1) shows that apart from controlling variability (see (4)), α also controls the number of ties which are expected in a sample. Let M denote the number of distinct values in a sample of n X_i s. We will refer to M as the number of clusters in the sample. Then, it is known that for large n , $E\{M | \alpha, n\} \approx \alpha \log(1 + n/\alpha)$ (Antoniak, 1974). As α tends to zero, most of the sampled X_i s share the same value, whereas when α tends to infinity, the X_i s will resemble a random sample from H_Θ . Therefore, the Dirichlet mixture model belongs to the family of finite mixture models (Titterington et al., 1985; Diebolt and Robert, 1994), with unknown number of components and a specific prior distribution on the number, the parameters and the weights of the components. This connection is investigated in Green and Richardson (2001).

2.1 Retrospective sampling from the Dirichlet process prior

This section introduces, in a very simple context, the technique of retrospective sampling and the two different computational approaches (marginal and conditional) to inference with Dirichlet mixture models. In particular, suppose that we wish to generate a sample $X = (X_1, \dots, X_n)$ from the Dirichlet process. There are essentially two different ways to achieve this. The first, obtains the sample by exploiting the Pólya urn scheme (10). In this method, X is simulated directly from its marginal distribution, with the random measure P being integrated out.

The second method uses a two-step hierarchical procedure. Initially, a realisation of P is simulated, and then n independent variables X_i are simulated from the generated P . The first step can, in principle, be implemented by simulating an infinite series of pairs (p_j, Z_j) , $j = 1, 2, \dots$, and forming P by the series representation (5). In order to draw n variables X_i conditionally on the simulated P , we first introduce a collection of classification variables $K = (K_1, \dots, K_n)$, one for each data point, where

$$K_i = j \text{ if and only if } X_i = Z_j, \tag{10}$$

therefore

$$\text{pr}\{K_i = j \mid p\} = p_j, \text{ for all } i = 1, \dots, n, j = 1, 2, \dots \quad (11)$$

Random generation of K_i , conditionally on the p_j s, can be done by first simulating U_i from a uniform distribution on $[0, 1]$, and then setting $K_i = j$ if and only if

$$\sum_{l=0}^{j-1} p_l < U_i \leq \sum_{l=1}^j p_l \quad (12)$$

where we define $p_0 = 0$ (Ripley, 1987). After K_i has been simulated, simply set

$$X_i = Z_{K_i}. \quad (13)$$

The merit of this conditional method lies in the fact that conditionally on P , the X_i s are independent and easy to simulate from, using (13). On the other hand, as described above, this method requires an infinite number of p_j s and Z_j s to be generated before simulating the classification variables K_i . This is clearly impossible, nevertheless, we show now how this can be avoided using the technique of *retrospective sampling*.

Retrospective sampling simply exchanges the order of simulation between the U_i and the pairs (p_j, Z_j) . If for a given U_i we need more p_j s than we currently have, in order to check (12), then we go back and simulate pairs (p_j, Z_j) *retrospectively*, until (12) is satisfied; the algorithm is as follows:

```

    Simulate  $p_1, Z_1$ , set  $N^* = 1$ ,  $i = 1$  and  $p_0 = 0$ 
  {
    1. Repeat until  $i > n$ 
    2. Simulate  $U_i \sim \text{Un}[0, 1]$ 
    3. if (12) is true for some  $k \leq N^*$  then
      {
         $K_i = k$ ,  $X_i = Z_k$ ,  $i = i + 1$ ,
        Goto 1
      }
    else,
      {
         $N^* = N^* + 1$ ,  $j = N^*$ , simulate  $p_j, Z_j$ 
        Goto 3
      }
  }

```

In this notation, N^* keeps track of how far into the infinite sequence $\{(p_j, Z_j), j = 1, 2, \dots\}$ we have visited. Notice that this algorithm serves as an informal proof of the Pólya urn representation of the sequence X_1, X_2, \dots , and shows that the two simulation approaches considered in this section are equivalent. Doss (1994) also points out that it is feasible to use (5) in order to simulate a sample X_1, \dots, X_n using a finite number of simulations.

These approaches correspond to two MCMC algorithms for fitting the Dirichlet mixture model to data $Y = (Y_1, \dots, Y_n)$. The MCMC algorithms, which are

not equivalent anymore, are presented in Sections 3 and 5. The *retrospective sampling* technique plays a key role in the implementation of the second algorithm. However, the details of the method for *a posteriori* sampling, given in Sections 5 and 6, are far more complex than the simple method given above.

3 Posterior inference and MCMC methods

When we fit (1) to data $Y = (Y_1, \dots, Y_n)$, there are a number of quantities we may want to make posterior inference about. These include i) the classification variables K_i , which can be used to classify the data into clusters, ii) the number of clusters M in the population, iii) the locations and the weights of the clusters in the population, $\{(p_j, Z_j) : K_i = j \text{ for at least one } i\}$, iv) the random measure P itself, or equivalently the infinite vector (p, Z) , v) the predictive distribution of future data, vi) the X_i s, and vii) the parameters α, Θ and ϕ . None of the existing methods can provide samples from the posterior distribution of P , without resorting to some kind of approximation, see for example Gelfand and Kottas (2002) for some suggestions. Notice however, that inference for the underlying measure P might not be the most interesting aspect of a semiparametric analysis using Dirichlet processes (Green and Richardson, 2001).

Analytic derivation of the corresponding posterior distributions is prohibited for Dirichlet mixture models, however Markov chain Monte Carlo (MCMC) techniques have been developed which allow sampling-based posterior inference. The standard MCMC method, which we term the marginal method, was introduced by Escobar (1994) and it was further explored in a number of papers, among which Escobar and West (1995), MacEachern (1994), Bush and MacEachern (1996) and Müller et al. (1996). Major improvements to the marginal approach were made by MacEachern and Müller (1998) and Neal (2000).

The marginal method is based on integrating out the random measure P . The Escobar and West (1995) Gibbs sampler, parameterises in terms of X and (Θ, α, ϕ) . The joint posterior distribution $\pi(X, \Theta, \alpha, \phi \mid Y)$ is not of known form, but the full conditional distributions

$$X_i \mid (Y, \Theta, \alpha, \phi, \{X_k, k \neq i\}), \quad i = 1, \dots, n$$

can be easily sampled from (Escobar, 1994) when H_Θ and $\pi(Y_i \mid X_i, \phi)$ form a conjugate pair, due to the Pólya urn representation (10). Thus, every sweep of the Escobar and West sampler first updates each of the X_i s in turn, and concludes by updating (usually in a single block) Θ, α and ϕ . One main drawback of the algorithm is that it becomes very difficult to implement when H_Θ and $\pi(Y_i \mid X_i, \phi)$ are not conjugate. This problem manifests itself when X_i is a random vector, as for example in (3), and we wish to assign independent priors for the different components of X_i . Another major weakness of the Escobar and West approach is the fact that a single-component Gibbs sampler is used to update each of the X_i , so that mixing of the MCMC can be very slow even for moderately sized data sets. Instead, we would like to update jointly all the X_i s which belong to the same cluster.

MacEachern and Müller (1998) describe a Gibbs sampler which can overcome both of the aforementioned problems. This method parameterises in terms of the allocation variables K_i defined in (10), the distinct values among the X_i s and (Θ, α, ϕ) , and integrates out the random weights $p = (p_1, p_2, \dots)$. The marginalisation over p has two main consequences. Firstly, integrating out p means that the different clusters are now exchangeable since the labeling of the clusters is arbitrary. Secondly, the allocation variables become *a priori* dependent, and their joint prior distribution can be derived by the Pólya urn representation (10). The exchangeability of the clusters is the key reason why this algorithm is simple to implement, even when H_Θ and $\pi(Y_i | X_i, \phi)$ are not conjugate.

On the other hand, the prior dependence among the K_i s can render their Gibbs sampler inefficient. For example, Neal (2000) points out one specific problem of the MacEachern and Müller algorithm, that it proposes new clusters less frequently than desirable. This can be of concern when modelling large datasets with many underlying clusters. For a thorough review and comparison of the different marginal algorithms see Neal (2000).

As a general point, received MCMC *wisdom* suggests that marginal samplers ought to be preferred to conditional ones. Some limited theory supports this view to some extent (see for example Liu (1994)). However it is common for marginalisation to destroy conditional independence structure which commonly assists the conditional sampler since conditionally independent components are effectively updated in one block. A striking example appears in Papaspiliopoulos et al. (2004), where Gibbs sampling using a marginal scheme requires $O(n)$ iterations to converge while a routine conditional method requires only $O(1)$ steps.

We note that Green and Richardson (2001) approached the computational problem from a different perspective and designed reversible jump MCMC algorithms for DMMs, but this is beyond the scope of our paper.

4 The conditional method

This section discusses an alternative computational approach for obtaining samples from the posterior distribution of quantities of interest (described in Section 3). The approach, which we call the conditional method, is related to the conditional method for simulating samples from the Dirichlet process prior (Section 2.1).

The main idea is, instead of integrating it out, to impute the random measure P and update it as part of the MCMC algorithm. Notice that imputing P is equivalent to imputing (p, Z) due to (5). It is also convenient to impute the classification variables K_i introduced in (10). Therefore, we re-write (1) using the following hierarchical structure,

$$\begin{aligned} Y_i | (Z, K, \phi) &\stackrel{iid}{\sim} \pi(Y_i | Z_{K_i}, \phi), \quad i = 1, \dots, n \\ K_i | p &\stackrel{iid}{\sim} \sum_{j=1}^{\infty} p_j \delta_j(\cdot) \end{aligned} \tag{14}$$

$$Z_j | \Theta \stackrel{iid}{\sim} \pi(Z_j | \Theta), j = 1, 2, \dots,$$

where $\pi(Z_j | \Theta)$ denotes the prior density of the Z_j s (i.e the density of $H_{\Theta}(\cdot)$ which for convenience we assume to exist). Moreover the prior distribution of the p_j s can be directly obtained from (7). Notice that (14) does not involve the X_i s explicitly, which instead are functions of the K_i s and Z (see (13)).

Having parameterised the model as in (14) the next step is to construct an MCMC algorithm which samples from the posterior distribution of $(K, p, Z, \Theta, \alpha, \phi)$ given observed data Y .

4.1 A “virtual” Gibbs sampler

Were we to have infinite computing capacity and storage available, the following algorithm would be a natural solution to the problem. We might obtain samples from the joint distribution of parameters and Dirichlet process by iterating the following four-component Gibbs sampler step:

1. Simulate from the distribution of Z conditionally on Y, K, Θ, ϕ .
2. Simulate from the distribution of p conditionally on K, α .
3. Simulate from the distribution of K conditionally on Y, p, Z, ϕ .
4. Simulate from the distribution of (α, Θ, ϕ) conditionally on Y, p, Z, K .
5. Goto 1.

Notice that the full conditional distributions in each of the steps above have been simplified due to conditional independences, for example in step 2, p conditionally on K and α is independent of all the other updated components. We shall defer giving the precise form of these conditional distributions for the time being, since this is the subject of Section 5.

Like the MacEachern and Müller algorithm, the above Gibbs sampler allows the joint updating of all the X_i s which share the same value. This is done in step 1, where Z is updated while keeping the allocation variables K fixed. On the other hand though, by imputing p , the K_i s become conditionally independent. Actually, we will see in Section 5 that in each of steps 1-3 the updated variables are conditionally independent. Thus the entire sampler is merely a 4 component Gibbs sampler (albeit on an infinite dimensional state space) no matter how large n is. Thus the algorithm’s mixing properties are potentially rather robust to the dimensionality of the problem; this is empirically supported by our simulation results in Section 6.1.

Unfortunately, the implementation of the above algorithm is prohibited by the fact that an infinite number of simulations has to be performed in steps 1 and 2, since both Z and p are countably infinite. Therefore, for example, updating Z entails simulating (and storing!) each of the Z_j s, $j = 1, 2, \dots$, from their full conditional distributions. Moreover, Section 5 shows that carrying out step 4 requires computing an infinite sum. This is why we term “virtual”

the above Gibbs sampler. Notice that the same difficulty is not encountered in the marginal method, since there the target distribution involves only a finite number of random variables (due to the exchangeability of the clusters).

4.2 Dirichlet process truncations

One approach to circumvent the problem of infinite simulations and storage requirements that the conditional method involves, was proposed by Ishwaran and Zarepour (2000). They approximate the infinite random measure P by a finite one P_N , such that P_N converges (in L^1) to P as $N \rightarrow \infty$. One of the possibilities is to truncate higher than the N th terms in the infinite sum representation of P in (5). Therefore,

$$P_N = \sum_{j=1}^N p_j \delta_{Z_j}(\cdot),$$

where the Z_j s and the p_j s are defined as in Section 2, but where it is set $V_N = 1$ in (6), to ensure that $\sum_{j=1}^N p_j = 1$. Clearly, as $N \rightarrow \infty$, P_N converges almost surely to P , and Ishwaran and Zarepour (2000) give some guidelines about sensible choices of N . Truncated Dirichlet process priors have also been used by Neal (1996).

Let $p_N = (p_1, \dots, p_N)$ and $Z_N = (Z_1, \dots, Z_N)$ to be the truncated versions of p, Z respectively. Ishwaran and Zarepour (2000) proceed by sampling from the posterior distribution $\pi(p_N, Z_N, K, \Theta, \alpha, \phi \mid Y)$ using the Gibbs sampler of Section 4.1, with p and Z replaced by their finite approximations p_N and Z_N correspondingly. Using simulated and real data examples, Ishwaran and Zarepour (2000) demonstrate that their method is more efficient and flexible than the Escobar and West sampler. Ishwaran and James (2001) make a comparison among three schemes, their conditional sampler, the Escobar and West algorithm and the Gibbs sampler suggested by MacEachern (1994). They again find that the Escobar and West scheme is the least efficient, whereas there is not a clear winner between the conditional scheme and the MacEachern algorithm. Nevertheless, it has to be mentioned that the most appropriate comparison is between the conditional method and the MacEachern and Müller algorithm (MacEachern and Müller, 1998), since the latter has been shown to be superior to the other marginal methods and since it is more similar in spirit to the conditional method.

The next section, taking the Ishwaran and Zarepour (2000) approach as a starting point, shows that the approximation of P by P_N is not necessary. In Section 6.1 we compare empirically how the conditional method compares with the MacEachern and Müller algorithm.

5 MCMC using retrospective sampling

It is obviously impossible to obtain a draw from the posterior distribution of the random measure P , since any such draw consists of infinitely many elements. If

the object of interest is P itself, or some functional of P , for example

$$\int_{\mathcal{X}} f(x)P(dx), \text{ for some real-valued function } f,$$

then it might be necessary to approximate P by a finite measure P_N and work with the latter instead. The results of Ishwaran and Zarepour (2000), Ishwaran and Zarepour (2002), and Gelfand and Kottas (2002) can be used to construct such an approximation.

On the other hand, this section demonstrates that, if the aim is to obtain samples from any of the posterior distributions described in Section 3 other than (iv), the approximation of P by P_N is unnecessary. Specifically, we show how the technique of retrospective sampling allows us to complete a sweep of the Gibbs sampler of Section 4 using only a finite number of simulations. The general idea is that at steps 1 and 2 only a small number of variates are simulated. If at the subsequent steps we need more p_j s and Z_j s than we have already simulated, we go back to steps 1 and 2 and simulate these values retrospectively. This is feasible due to the conditional independence structure in (14). We now give a detailed description of the method.

For ease of exposition, this section assumes that the hyperparameters ϕ, Θ and α are considered known. Hence, we illustrate how to iterate only steps 1 to 4 of the algorithm given in Section 4.1. Modifications of our approach when the hyperparameters are unknown will be discussed in Section 7.

We begin by introducing some terminology and notation. For a given configuration of the classification variables K_1, \dots, K_n , we define

$$m_j = \sum_{i=1}^n \mathbb{1}[K_i = j], \quad j = 1, 2, \dots, \quad (15)$$

which is the number of data points allocated to the j th component in (5); clearly $0 \leq m_j \leq n$ for all $j = 1, 2, \dots$, and $\sum_j m_j = n$. Moreover, let

$$\begin{aligned} I &= \{1, 2, \dots\} \\ I^{(al)} &= \{j \in I : m_j > 0\} \\ I^{(d)} &= \{j \in I : m_j = 0\} = I - I^{(al)}. \end{aligned} \quad (16)$$

I is the set of all positive integers representing all components in the infinite mixture (5). For a given configuration of K_1, \dots, K_n , $I^{(al)}$ is the set of all *alive* components, in the sense that some data points have been allocated to them ($m_j > 0$), whereas $I^{(d)}$ is the set of remaining components, which we refer to as *dead*.

We now derive the conditional distributions in each of the steps 1-4 of the Gibbs sampler in Section 4.1, and show how simulation in these steps can be achieved. It is easy to check that,

$$Z_j \mid (Y, K, \Theta, \phi) \sim \begin{cases} H_{\Theta}, & \text{for all } j \in I^{(d)} \\ \prod_{i:K_i=j} \pi(Y_i \mid Z_j, \phi) \pi(Z_j \mid \Theta) & \text{for all } j \in I^{(al)} \end{cases} \quad (17)$$

and that Z_j is *a posteriori* independent of Z_k for any $j \neq k$. Therefore, it is easy to simulate from Z_j for all $j \in I$. If $j \in I^{(d)}$ then simply simulate Z_j from the prior H_Θ . When H_Θ and $\pi(Y_i | Z_j, \phi)$ form a conjugate pair (as in (2)), the conditional distribution of Z_j is of known form for all $j \in I^{(al)}$, and thus can be sampled easily. More generally, adaptive rejection sampling (Wild and Gilks, 1993) or a Metropolis-Hastings step can be used to update $Z_j, j \in I^{(al)}$. In particular, when Z_j is partitioned in components, as in (3) where $Z_j = (\mu_j, \sigma_j^2)$, a Gibbs sampler step can be used to update each component conditional on the rest.

It can be also be shown that

$$V_j | K, \alpha \sim \text{Be}(m_j + 1, n - \sum_{l=1}^j m_l + \alpha) \text{ for all } j = 1, 2, \dots, \quad (18)$$

and that V_j is *a posteriori* independent of V_k for all $j \neq l$. Simulation of the p_j s in step 3 of the Gibbs sampler needs some care. If a draw from p_j for some $j \geq 1$ is required, we recommend first simulating (and storing) V_1, \dots, V_j from (18), and then applying (6). If another draw p_h is required then: (i) if $h < j$, apply (6) using the simulated $V_l, l = 1, \dots, h$, (ii) if $h > j$, simulate (and store) V_{j+1}, \dots, V_h and apply (6).

Assume that in steps 1 and 2 we have simulated, as described earlier, p_j, Z_j only for $j \leq N^*$, where $N^* \geq 1$ and its choice will be described later in this section. The conditional distribution of K in step 3 of the Gibbs sampler in Section 4.1 is characterised by the fact that

$$\text{pr}\{K_i = j | Y, p, Z, \phi\} \propto p_j \pi(Y_i | Z_j, \phi), \quad j = 1, 2, \dots, \quad (19)$$

and the independence of K_i and K_m for all $i \neq m$. The normalising constant of (19) is

$$c_i = \sum_{j=1}^{\infty} p_j \pi(Y_i | Z_j, \phi). \quad (20)$$

However, direct simulation of K_i requires computation of c_i , which involves computing an infinite sum. Notice that in the MacEachern and Müller algorithm the K_i s are dependent conditionally on everything else. On the other hand, in their algorithm it is easy to compute the normalising constant for the conditional distribution of each allocation variable K_i . This is due to the fact that the clusters are exchangeable. Roughly speaking, when we propose to allocate data point i to a new cluster, it does not matter to which cluster it will be allocated to. In the conditional augmentation, the clusters are not exchangeable since the p_j s differ, thus the posterior probability of being allocated to one of the infinitely many currently dead clusters j depends on j , consequently c_i involves an infinite sum.

Using retrospective sampling, we describe a simple method for updating the allocation variables, while using only a finite number of simulations ($N^* < \infty$) at steps 1 and 2 of the Gibbs sampler and avoiding computing the infinite sum (20).

6 A quasi-independence sampler for the allocation variables

One simple and effective way to avoid the computation of the normalising constant c_i , is to update the allocation variables using a Metropolis-Hastings step. We propose an algorithm which by construction has acceptance probability 1 for many “interesting” proposed values, and it resembles an independence sampler.

Let $k = (k_1, \dots, k_n)$ denote the current configuration of $K = (K_1, \dots, K_n)$, i.e the current values of the K_i s before entering step 3 of the Gibbs sampler. Let

$$\max\{k\} = \max_i k_i \quad (21)$$

be the maximal element of the vector k , and

$$k(i, j) = (k_1, \dots, k_{i-1}, j, k_{i+1}, \dots, k_n)$$

be the vector produced from k by substituting the i th element by j . Our Metropolis-Hastings algorithm updates each of the K_i s in turn (note that the K_i s are conditionally independent). When updating K_i , the sampler proposes to move from k to $k(i, j)$, $j \in I$. The distribution which the proposed value j is generated from, depends upon the current configuration k only through $\max\{k\}$, and takes the form

$$q_i(k, j) \propto \begin{cases} p_j \pi(Y_i | Z_j, \phi), & \text{for } j \leq \max\{k\} \\ M_i p_j, & \text{for } j > \max\{k\} \end{cases} \quad (22)$$

with normalising constant

$$c_i(\max\{k\}) = \sum_{j=1}^{\max\{k\}} p_j \pi(Y_i | Z_j, \phi) + M_i (1 - \sum_{j=1}^{\max\{k\}} p_j), \quad (23)$$

which can be easily computed given $\{(p_j, Z_j) : j \leq \max\{k\}\}$. Notice that, conditional on being less than $\max\{k\}$, j is proposed from a probability mass function proportional to the actual conditional posterior distribution (19), whereas conditionally on being larger than $\max\{k\}$, j is proposed from the prior distribution. The constant M controls the probability of proposing j greater than $\max\{k\}$, which is

$$(1 - \sum_{j=1}^{\max\{k\}} p_j) M_i / c_i(\max\{k\}). \quad (24)$$

A conservative choice, which however guarantees good theoretical properties of the sampler (see below), is to choose M_i so that the probability of proposing j greater than $\max\{k\}$ is greater than the prior probability assigned to the set $\{j : j > \max\{k\}\}$. Therefore, we want to choose M_i so that

$$\sum_{j=1}^{\max\{k\}} p_j \pi(Y_i | Z_j, \phi) + M_i (1 - \sum_{j=1}^{\max\{k\}} p_j) \leq M_i. \quad (25)$$

For this inequality to be true, M_i typically will need to depend on $\max\{k\}$ and possibly on $\phi, \{Z_j, j \leq \max\{k\}\}$, and Y_i , but to avoid notation overload we will simply write $M_i(\max\{k\})$. A choice which satisfies (25) and which will use in our examples, is

$$M_i(\max\{k\}) = \max\{\pi(Y_i | Z_j, \phi), j \leq \max\{k\}\}. \quad (26)$$

A careful calculation yields that the Metropolis-Hastings acceptance probability of the transition from $K = k$ to $K = k(i, j)$ is

$$\alpha_i(k, k(i, j)) = \begin{cases} 1, & \text{if } j \leq \max\{k\} \text{ and } \max\{k(i, j)\} = \max\{k\} \\ \min \left\{ 1, \frac{c_i(\max\{k\})}{c_i(\max\{k(i, j)\})} \frac{M(k(i, j))}{\pi(Y_i | Z_{k(i, j)}, \phi)} \right\}, & \text{if } j \leq \max\{k\} \text{ and } \max\{k(i, j)\} < \max\{k\} \\ \min \left\{ 1, \frac{c_i(\max\{k\})}{c_i(\max\{k(i, j)\})} \frac{\pi(Y_i | Z_j, \phi)}{M(k)} \right\}, & \text{if } j > \max\{k\}. \end{cases}$$

Thus, our algorithm behaves like a Gibbs sampler when moves to $j \leq \max\{k\}$ are proposed, since such moves are accepted as long as the normalising constant (23) is not affected by the transition from k to $k(i, j)$ (which corresponds to the condition $\max\{k(i, j)\} = \max\{k\}$). If the proposed transition changes the normalising constant (thus $\max\{k(i, j)\} \neq \max\{k\}$), (23) needs to be evaluated for the proposed state of the classification variables $k(i, j)$, and the acceptance probability has to be computed.

The updating of the K_i s using the above MCMC scheme is done using retrospective sampling. Before entering step 3 of the Gibbs sampler (see Section 4.1), we have only simulated $\{(p_j, Z_j) : j \leq \max\{k\}\}$, and we set $N^* = \max\{k\}$. For each $i = 1, \dots, n$, we simulate $U_i \sim \text{Un}[0, 1]$ and propose to set $K_i = j$, where j satisfies

$$\sum_{l=0}^{j-1} q_i(k, l) < U_i \leq \sum_{l=1}^j q_i(k, l), \quad (27)$$

with $q_i(k, 0) \equiv 0$. If (27) is not satisfied for any $j \leq N^*$, then we increase $N^* = N^* + 1$, we go back to steps 1 and 2 and simulate (and store) a pair (p_{N^*}, Z_{N^*}) retrospectively until (27) is true. The proposed value is then accepted with probability $\alpha_i(\max\{k\}, j)$. Therefore, steps 1-3 of the retrospective algorithm are implemented as follows:

- Give initial allocation $k = (k_1, \dots, k_n)$, set $N^* = \max\{k\}$
1. Simulate $Z_j, j \leq \max\{k\}$ from its conditional posterior.
 2. Simulate $p_j, j \leq \max\{k\}$ from its conditional posterior.
 - 3.1 Repeat until $i > n$
 - 3.2. simulate $U_i \sim \text{Un}[0, 1]$
 - 3.3 if (27) is true for some $j \leq N^*$ then
 - {
 - propose to move to $K_i = j$
 - accept the move with probability $\alpha_i(k, k(i, j))$

```

    goto 3.1
}
else,
{
     $N^* = N^* + 1, j = N^*$ 
    simulate  $(p_j, Z_j)$  retrospectively from their prior
    goto 3.3.
}
Set  $N^* = \max\{k\}$ 

```

Notice that both $\max\{k\}$ and N^* are changing during the updating of the K_i s. Thus, the proposal distribution (22) is adapting itself to improve approximation of the target distribution (19). Specifically, at the early stages of the algorithm N^* will be large, but as the cluster structure starts being identified by the data then N^* will take much smaller values. Nevertheless, the adaptation of the algorithm does not violate the Markov property of the chain with stationary distribution the joint posterior distribution of (p, Z, K) . It is recommended to update the K_i s in a random order, to avoid using systematically less efficient proposals for some of the variables.

Another desirable characteristic of the proposal distribution (22) is that it has heavier tails than the target distribution (19), when M is chosen as in (26). This property has been shown to be necessary and sufficient for independence MCMC samplers in order to be uniformly ergodic (Mengersen and Tweedie, 1996).

6.1 A brief simulation study

In order to assess the efficiency of the sampler we have tested it on various simulated datasets of size 100 suggested by Green and Richardson (2001). We report results for the “lepto” dataset, simulated from the unimodal leptokurtic mixture, $0.67N(0, 1) + 0.33N(0.3, 0.25^2)$, and for the “bimod” dataset, simulated from the bimodal mixture, $0.5N(-1, 0.5^2) + 0.5N(1, 0.5^2)$; see Section 3.2 of Green and Richardson (2001) for more details on these datasets. We fit model (3), with fixed hyperparameters, chosen in a data-driven way as suggested by Green and Richardson (2001). Specifically, let R be the range of the data. We set $\mu = R/2, \sigma_z = R, \gamma = 2, \beta = 0.02R^2$ and we consider several different values of $\alpha = 0.1, 0.3, 0.5, 1, 2$. Data-dependent choice of hyperparameters is commonly made in mixture models, see for example Richardson and Green (1997). We use a Gibbs sampler to update $Z_j = (Z_j^{(1)}, Z_j^{(2)})$ for every j in step 1 of the algorithm, i.e we update $Z_j^{(2)}$ given $Z_j^{(1)}$ and the rest, and then $Z_j^{(1)}$ given the new value of $Z_j^{(2)}$ and the rest.

We monitor two functionals of the updated variables, the number of alive components, M , and the deviance of the estimated density. Following Green and Richardson (2001), if h is an estimate of the density of data $Y = \{Y_1, \dots, Y_n\}$,

the deviance is defined as

$$D(h) = -2 \sum_{i=1}^n \log\{h(Y_i)\}.$$

In a Bayesian framework, the density estimate is the posterior predictive density of Y_{n+1} , namely

$$\pi(Y_{n+1} | Y) = E\{\pi(Y_{n+1} | X_1, \dots, X_{n+1}, \phi) | Y\}$$

where the expectation is with respect to the conditional distribution of $\{X_1, \dots, X_{n+1}\}$ given Y . When α is small compared to n , the predictive density can be approximated by

$$\pi(Y_{n+1} | Y) \approx E\{h(Y_{n+1} | \{Z_j, j \in I^{(al)}\}) | Y\},$$

where the expectation is with respect to the conditional distribution of $\{Z_j, j \in I^{(al)}\}$ given Y , and h takes the kernel density estimate form

$$h(Y_{n+1} | \{Z_j, j \in I^{(al)}\}) = \sum_{j \in I^{(al)}} \frac{m_j}{n} \pi(Y_{n+1} | Z_j, \phi); \quad (28)$$

see for example Section 2 of Müller et al. (1996). At every iteration of the algorithm we compute $h(Y_i | \{Z_j, j \in I^{(al)}\})$ for $i = 1, \dots, n$, and we evaluate and store $D(h)$. The deviance is chosen as a meaningful function of all parameters, which also gives an indication on how well the algorithm explores the high-dimensional model space. The efficiency of the sampler is summarised by reporting for M and $D(h)$ the estimated integrated autocorrelation time, $\tau = 1 + 2 \sum_{j=1}^{\infty} \rho_j$, where ρ_j is the lag- j autocorrelation of the monitored chain. This is a standard way of measuring the speed of convergence of square integrable functions of an ergodic Markov chain (see for example Roberts (1996)), which has also been used by Green and Richardson (2001) and Neal (2000) in the context of DMMs recently. However, consistent estimation of τ is non-trivial and we have used the batch-means approach, discussed for example in Roberts (1996)

The algorithm was run for 130,000 iterations, the first 30,000 were discarded as a burn-in, and the rest were thinned 1/20. The estimated τ s for various values of α are given in Table 1. It can be seen that the algorithm is generally mixing rapidly, and the algorithmic performance is particularly good for large values of α . The mixing is slightly worse for the “lepto” dataset as the posterior distribution is uncertain about the existence of the smaller Gaussian component, leading to some degree of bimodality. For the “lepto” dataset, mixing is worst for very small values of α , where there is a considerable prior/likelihood conflict, leading to a severe bimodality in the posterior distribution as illustrated by Figure 1. In general, our results are comparable with those of the MacEachern and Müller marginal algorithm, see for example Section 5.2 of Green and Richardson (2001). In the conditional approach, it is the dependence between K and (p, Z) which affects the mixing of the algorithm, and this clearly depends on α . This is reflected in the τ s decreasing for larger values of α .

α	M		D(h)	
	lepto	bimod	lepto	bimod
0.1	25.42	6.58	28.32	0.78
0.3	11.10	5.7	10.0	0.70
0.5	5.25	4.22	3.68	1.07
1	3.04	3.58	1.95	0.83
2	2.12	3.02	1.19	1.26

Table 1: Integrated autocorrelation times for number of clusters M and deviance $D(h)$, for various values of α , for the “lepto” and “bimod” datasets.

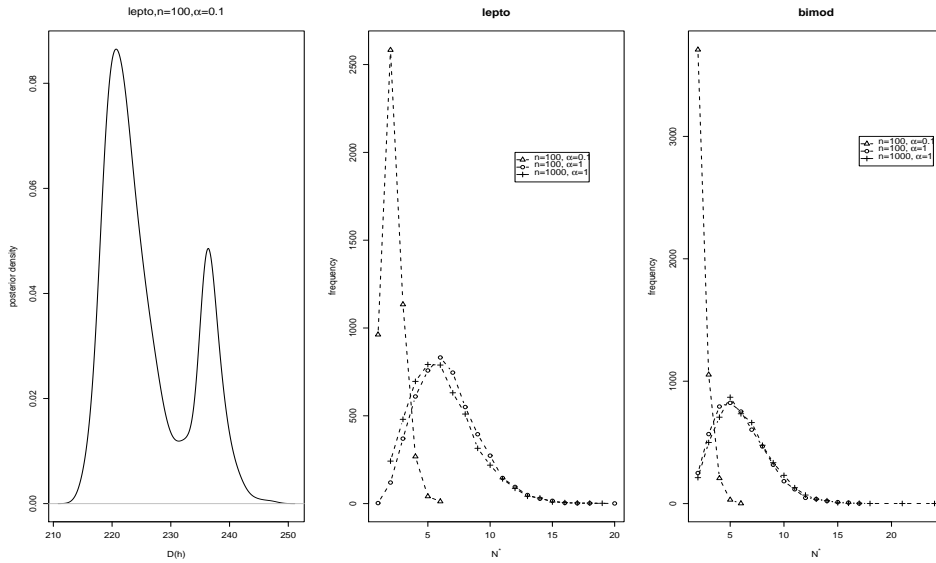


Figure 1: Left: posterior distribution of $D(h)$ for the “lepto” dataset and $n = 100, \alpha = 0.1$. Middle and right: frequencies for different values that N^* took, and obtained after removal of burn-in, for $n = 100$ and $\alpha = 0.1, 1$ and $n = 1000, \alpha = 1$, for the “bimod” (middle) and the “lepto” (right) datasets. Notice that the maximum value of N^* obtained was for the “lepto” (resp. “bimod”) example: 6 (resp. 6) for $\alpha = 0.1$, 20 (resp. 17) for $n = 100, \alpha = 1$ and 19 (resp. 22) for $n = 1000, \alpha = 1$.

The fact that the parameters are grouped in 3 blocks of conditionally independent variables suggests that the conditional MCMC algorithm might be well suited for fitting large datasets. We simulate a further 900 data from the leptokurtic and the bimodal mixtures and we fit model (3) on the 1000 simulated points. All data are initially allocated to a single cluster with parameter values simulated from the prior. Hyperparameters are chosen as described earlier and we take $\alpha = 1$. Figure 2 shows results for the “bimod” dataset, but the picture is the same for the “lepto” dataset as well. The performance of the algorithm is very good, and it quickly escapes from the initial state where all data are allocated to the same cluster (with parameter values drawn from the prior) and then it mixes well. For the “lepto” and “bimod” datasets respectively, the esti-

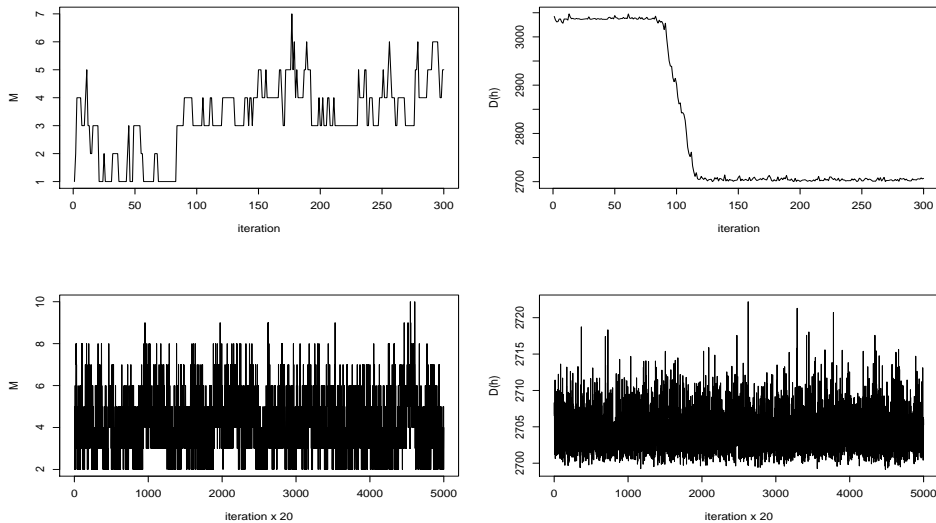


Figure 2: Top: The first 300 iterations of the algorithm for $n = 1000$. Bottom: Sample paths for M and $D(h)$ when the algorithm has converged (chains have been thinned $1/20$).

mated integrated autocorrelation times for M are 11.87 and 11.6, and for $D(h)$ they are 1.83 and 1.08. These results support our intuition and the heuristic arguments of Sections 3 and 4.1 that the conditional algorithm is well suited for fitting large datasets.

As shown in Figure 1, N^* typically takes small values during the iterations of the algorithm. Interestingly, N^* depends crucially on α , but not on the size of the fitted dataset.

7 Inference and non-centred parametrisations for hyper-parameters

In hierarchical models involving hidden stochastic processes, inference for the (hyper)parameters of the hidden process is desirable because they are often associated with certain population characteristics. Moreover we will demonstrate by example that inference for the hidden process itself might be improved by updating the parameters as well.

In principle, updating the hyperparameters is straightforward in an MCMC algorithm. In particular, in the Gibbs sampler of Section 4.1 it comes down to implementing step 4 as well as steps 1-3. Notice, however, that due to the hierarchical structure, (α, Θ) are independent of (Y, K) conditionally upon (p, Z) (see (14)). Less crucially, α and Θ are also conditionally independent given (p, Z) . However, (p, Z) contain an infinite amount of information about (α, Θ) . As we showed in Section 5, steps 1,2 of the Gibbs sampler simulate (in principle) an infinite number of independent $V_j \sim \text{Be}(1, \alpha)$, and $Z_j \sim H_\Theta$. Therefore, the Gibbs sampler described in Section 4.1 becomes reducible when the hyperparameters are updated. Notice that this is an inherent problem of the conditional approach, which is not really circumvented by approximating the Dirichlet process by a finite measure, as advocated by Ishwaran and Zarepour (2000) and it is reviewed in Section 4.2. When the Dirichlet measure is approximated by a finite measure, the conditional algorithm is not reducible, but its convergence gets arbitrarily bad as the approximation gets arbitrarily accurate; see Roberts and Stramer (2001) for a similar problem in the context of inference for partially observed diffusions.

This type of convergence problems are very common when MCMC is used to infer about hierarchical models with hidden stochastic processes, see Papaspiliopoulos et al. (2003) for a review. It is often the case that there is a strong prior dependence between the hidden process and its parameters. Consequently, the information about the hyperparameters contained in the imputed process is much greater than that contained in the observed data, and the Gibbs sampler which updates successively the parameters and the hidden process converges slowly. In the conditional algorithm for the DMMs this problem is exaggerated since the parameters are uniquely determined by the imputed process.

However, there is an established methodology for tackling such problems. Papaspiliopoulos (2003), Roberts et al. (2003) and Papaspiliopoulos et al. (2003) suggest a so-called *non-centred reparametrisation* of the hierarchical model under which the hidden process and the parameters are apriori independent. This parametrisation is in contrast with the more “natural” *centred* parametrisation, under which the hyperparameters are conditionally independent of the rest, given the the process. The Gibbs sampler under a non-centred parametrisation has often much faster convergence rate than under a centred parametrisation (Papaspiliopoulos et al., 2003), but for the model considered in this paper it is also the only choice, since the algorithm is reducible otherwise.

Therefore, we seek random variables \tilde{V}_j and \tilde{Z}_j , which are apriori inde-

pendent of (α, Θ) , and transformations h_1, h_2 , such that $V_j \stackrel{d}{=} h_1(\tilde{V}_j, \alpha)$ and $Z_j \stackrel{d}{=} h_2(\tilde{Z}_j, \Theta)$ ($\stackrel{d}{=}$ denotes equality in distribution). H_Θ typically belongs in some simple parametric family, and Papaspiliopoulos (2003) and Papaspiliopoulos et al. (2003) contain several suggestions for finding \tilde{Z}_j . For example, when $Z_j \sim N(\mu, \sigma_z^2)$, as in (2), we can take $\tilde{Z}_j \sim N(0, 1)$, and $h_2(\tilde{Z}_j, \mu, \sigma_z) = \sigma_z \tilde{Z}_j + \mu$. However, for the rest of the paper we will focus on inference about α , for which no reasonable data-dependent value can be chosen a priori, and whose value critically affects the number of alive components (see Section 2). On the contrary Θ will be treated as fixed. It is feasible to find a non-centred reparametrisation of (V_j, α) exploiting the inverse CDF method for generating random variables. It is to check that if $\tilde{V}_j \sim \text{Un}(0, 1)$, then

$$V_j \stackrel{d}{=} 1 - \tilde{V}_j^{1/\alpha}.$$

The Gibbs algorithm under the non-centred parametrisation is a trivial extension of the algorithm presented in Section 6. Steps 1-3 of the algorithm proceed as already described, since it is important to note that when we condition upon α , updating the V_j s or the \tilde{V}_j s is equivalent. Thus we will now describe how to implement step 4 of the algorithm. Let $\tilde{V} = (\tilde{V}_1, \tilde{V}_2, \dots)$. Notice that conditionally upon K and \tilde{V} , α is independent of the rest, and this distribution is given by

$$\pi(\alpha \mid K, \tilde{V}) \propto \prod_{j \in I^{(a)}} p_j(\alpha, \tilde{V})^{m_j} \pi(\alpha), \quad (29)$$

where $\pi(\alpha)$ is the prior distribution of α . As suggested by the notation, under this parametrisation p is now a function of both \tilde{V} and α , since

$$p_1(\alpha, \tilde{V}) = 1 - \tilde{V}_1^{1/\alpha}, \quad p_j(\alpha, \tilde{V}) = \left\{ \prod_{t=1}^{j-1} \tilde{V}_t \right\}^{1/\alpha} (1 - \tilde{V}_j^{1/\alpha}). \quad (30)$$

(29) is not of known form, however it can be easily computed for any value of α , thus we use a Metropolis-Hastings step to update α . Therefore, we propose according to some proposal distribution a new value, say $\hat{\alpha}$, and the probability of accepting this transition depends on the ratio of the posterior densities for the current and proposed values, which under some algebra it can be shown to take the simple form:

$$\begin{aligned} \log \left\{ \frac{\pi(\hat{\alpha} \mid K, \tilde{V})}{\pi(\alpha \mid K, \tilde{V})} \right\} &= m_1 (\log\{1 - V_1^{\alpha/\hat{\alpha}}\}) \\ &+ \sum_{j=2}^{\max\{k\}} m_j \left((\alpha/\hat{\alpha} - 1) \log p_j - (\alpha/\hat{\alpha}) \log V_j + \log\{(1 - V_j)^{\alpha/\hat{\alpha}}\} \right); \end{aligned}$$

in the above expression, $V_j = 1 - \tilde{V}_j^{1/\alpha}$ for all j .

In comparison to the case where α is fixed, the inference about the number of alive clusters is significantly improved by updating α rather than fixing it to a predetermined value. This is illustrated in Figure 3, where we compare the posterior distribution of the number of alive clusters for the “lepto” and the “bimod” datasets (for $n = 100$ and $n = 1000$), when α is kept fixed to $\alpha = 1$ (which suggests that we expect a priori a moderate number of clusters), and when α is updated (using an exponential prior with mean 10). When α is updated there is much more support for the “correct” answer, that the number of clusters is 2, in both cases.

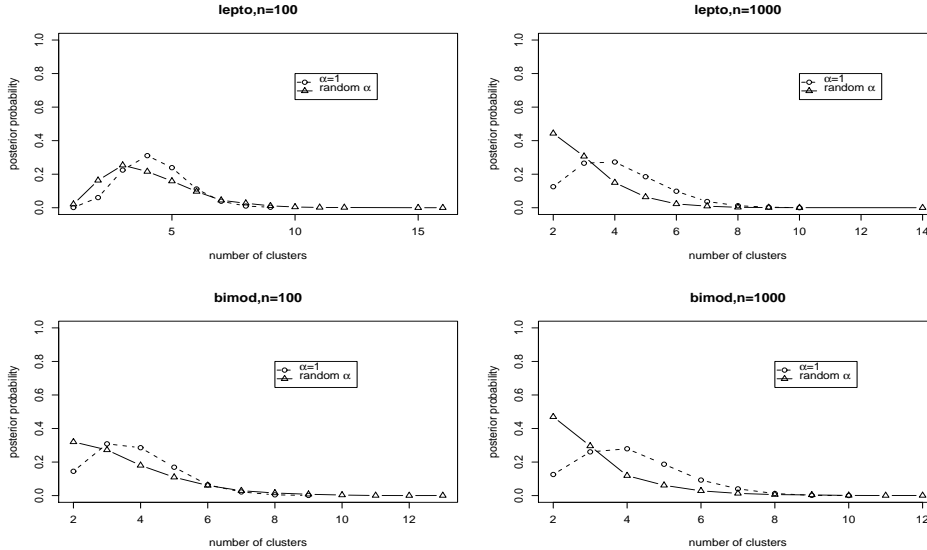


Figure 3: The posterior distribution of the number of alive clusters.

Figure 4 shows a kernel density estimate of the posterior density of $\log \alpha$, for the “bimod” dataset, under two different priors, an exponential with mean 10, and the much less dispersed exponential with mean 1. The picture is similar for the “lepto” dataset. Figure 5 shows 5 realisation from the posterior distribution of the density estimate (28) (for $n = 1000$ and α being updated with prior an exponential with mean 10), computed at a fine grid, together with the true “bimod” density that the data were generated from, and the kernel density estimate using Silverman’s suggestion for the choice of bandwidth. It is interesting that the estimates from the DMM are doing much better than the kernel density estimate.

8 Discussion

In this paper we have introduced a new algorithm for performing fully Bayesian inference for Dirichlet models. By using the conditional specification of the

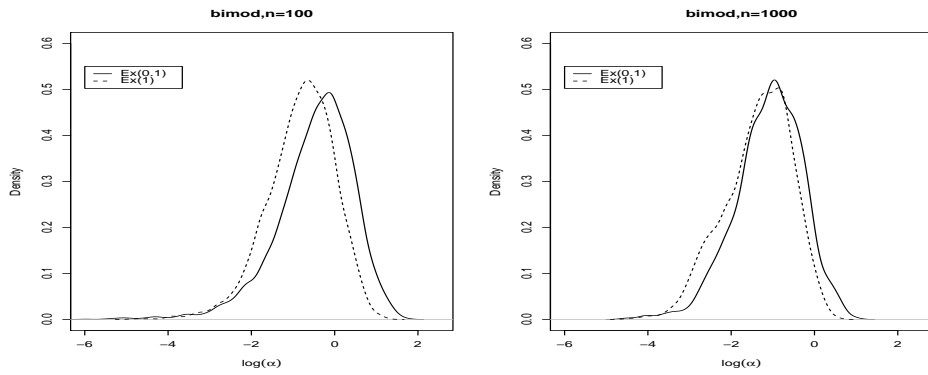


Figure 4: The posterior distribution of $\log \alpha$ for the “bimod” dataset, under two different priors.

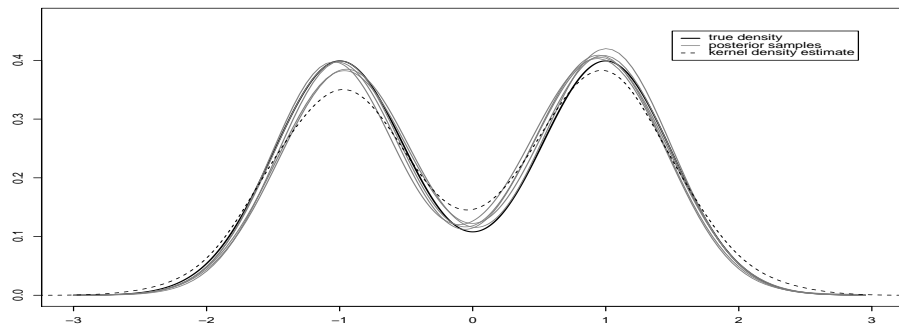


Figure 5: The “bimod” density (solid black), together with 5 realisations from the posterior distribution of the density estimate (28) (solid gray) and the kernel density estimate (dashed).

model, we obtain algorithms which are relatively robust to large datasets, and by using retrospective methodology we do not require any approximation to the Dirichlet process. Thus our methods have been successfully implemented for large datasets, and all our algorithms generally give well-mixing Markov chains. There is one exception to this general pattern: for fixed and very small α , in the “lepto” example, mixing is poor. Here however we saw that this was due to a marked bimodality which was caused by a strong prior/likelihood conflict.

Hyperparameter updating requires non-centering methodology to avoid reducibility. Our methods prove to be fairly easy to implement and computationally inexpensive. Our results also suggest that improved identification of the number of clusters can be achieved by allowing α to vary.

In extending this work, we have already managed to design an exact Gibbs sampler, ie one in which in which the allocation variables are simulated directly from their conditional posterior distributions. This is carried out by an intricate coupling of the Dirichlet process which permits tight bounds on c_i as defined in (20) and also allows retrospective simulation of all related variables. Although the resulting algorithm is simple to implement, the mathematical construction behind this method is very involved and will be shortly reported elsewhere.

Retrospective sampling is a methodology of great potential for other problems involving Bayesian non- and semi-parametric modelling. Currently, we are developing methods for exact simulation of diffusions (Beskos and Roberts, 2004) with a view to performing exact Bayesian inference from partially observed diffusions. This work and other extensions of the current paper will also be reported elsewhere.

Acknowledgments

We would like to thank Martin Sköld and Alex Beskos for motivating discussions, and Peter Green for helpful suggestions.

References

- Antoniak, C. E. (1974) Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, **2**, 1152–1174.
- Beskos, A. and Roberts, G. (2004) Exact simulation of diffusions. *submitted*.
- Blackwell, D. and MacQueen, J. B. (1973) Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, **1**, 353–355.
- Bush, C. A. and MacEachern, S. N. (1996) A semiparametric Bayesian model for randomised block designs. *Biometrika*, **83**, 275–285.
- Diebolt, J. and Robert, C. (1994) Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society, Series B*, **56**, 363–375.

- Doss, H. (1994) Bayesian nonparametric estimation for incomplete data via successive substitution sampling. *The Annals of Statistics*, **22**, 1763–1786.
- Escobar, M. D. (1994) Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, **89**, 268–277.
- Escobar, M. D. and West, M. (1995) Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, **90**, 577–588.
- Ferguson, T. S. (1973) A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, **1**, 209–230.
- (1983) Bayesian density estimation by mixtures of normal distributions. In *Recent Advances in Statistics: Papers in Honor of Herman Chernoff on His Sixtieth Birthday*, 287–302. Academic (New York; London).
- Gelfand, A. and Kottas, A. (2003) Bayesian Semiparametric Regression for Median Residual Life. *Scand. J. Statist.*, **30**, 651–665.
- Gelfand, A. E. and Kottas, A. (2002) A computational approach for full nonparametric Bayesian inference under Dirichlet process mixture models. *J. Comput. Graph. Statist.*, **11**, 289–305.
- Green, P. and Richardson, S. (2001) Modelling Heterogeneity With and Without the Dirichlet Process. *Scand. J. Statist.*, **28**, 355–375.
- Ishwaran, H. and James, L. (2001) Gibbs Sampling Methods for Stick-Breaking Priors. *Journal of the American Statistical Association*, **96**, 161–173.
- Ishwaran, H. and Zarepour, M. (2000) Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika*, **87**, 371–390.
- (2002) Exact and approximate sum-representations for the dirichlet process. *Canadian J. Statistics*, **30**, 269–283.
- 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**, 958–966.
- MacEachern, S. and Müller, P. (1998) Estimating Mixture of Dirichlet Process Models. *Journal of Computational and Graphical Statistics*, **7**, 223–238.
- MacEachern, S. N. (1994) Estimating normal means with a conjugate style Dirichlet process prior. *Communications in Statistics, Part B – Simulation and Computation*, **23**, 727–741.
- Mengersen, K. L. and Tweedie, R. L. (1996) Rates of convergence of the Hastings and Metropolis algorithms. *Ann. Statist.*, **24**, 101–121.

- Müller, P., Erkanli, A. and West, M. (1996) Bayesian curve fitting using multivariate normal mixtures. *Biometrika*, **83**, 67–79.
- Neal, R. (2000) Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, **9**, 283–297.
- Neal, R. M. (1996) *Bayesian Learning for Neural Networks*. Springer-Verlag.
- Papaspiliopoulos, O. (2003) Non-centered parameterisations for hierarchical models and data augmentation. *PhD Dissertation*, Department of Mathematics and Statistics, Lancaster University, Lancaster.
- Papaspiliopoulos, O., Roberts, G. O. and Sköld, M. (2003) Non-centered parameterisations for hierarchical models and data augmentation. In *Bayesian Statistics 7* (eds. J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith and M. West), 307–327. Oxford: Oxford University Press.
- Papaspiliopoulos, O., Roberts, G. O. and Sköld, M. (2004) A comparison of non-centering and marginalisation for mcmc algorithms. *in preparation*.
- Petrone, S. and Raftery, A. E. (1997) A note on the Dirichlet process prior in Bayesian nonparametric inference with partial exchangeability. *Statistics & Probability Letters*, **36**, 69–83.
- Quintana, F. and Iglesias, P. (2003) Bayesian clustering and product partition models. *Journal of the Royal Statistical Society, Series B, Methodological*, **65**, 557–574.
- Richardson, S. and Green, P. J. (1997) On Bayesian analysis of mixtures with an unknown number of components (with discussion). *J. R. Statist. Soc. B*, **59**, 731–792.
- Ripley, B. D. (1987) *Stochastic Simulation*. Wiley.
- Roberts, G. and Stramer, O. (2001) Bayesian inference for incomplete observations of diffusion processes. *Biometrika*, **88**, 203–221.
- Roberts, G. O. (1996) Markov chain concepts related to sampling algorithms. In *MCMC in practice* (eds. W. Gilks, S. Richardson and D. Spiegelhalter), 45–57. Chapman and Hall.
- Roberts, G. O., Papaspiliopoulos, O. and Dellaportas, P. (2003) Bayesian inference for Non-Gaussian Ornstein-Uhlenbeck Stochastic Volatility processes. *J. R. Statistic. Soc. B*, **66**, 369–394.
- Sethuraman, J. (1994) A constructive definition of Dirichlet priors. *Statistica Sinica*, **4**, 639–650.
- Titterton, D. M., Smith, A. F. M. and Makov, U. E. (1985) *Statistical Analysis of Finite Mixture Distributions*. Wiley.

Walker, S., Damien, P., Laud, W. and Smith, A. (1999) Bayesian nonparametric inference for random distributions and related functions (with discussion). *Journal of the Royal Statistical Society, Series B, Methodological*, **61**, 485–527.

Wild, P. and Gilks, W. R. (1993) Algorithm As 287: Adaptive rejection sampling from log-concave density functions. *Appl. Statist.*, **42**, 701–708.