# Filtering Methods for Mixture Models

Paul Fearnhead and Loukia Meligkotsidou
Department of Mathematics and Statistics
Lancaster University

## Summary

We consider Bayesian inference for mixture distributions of known number of components via a set of filtering recursions. We extend the method of direct simulation for discrete mixture distributions of Fearnhead (2005) in order to analyse continuous mixture models. Furthermore, we introduce resampling steps similar to those in particle filters within the steps of the filtering recursions, which make calculations efficient and enable us to analyse larger data sets. The proposed algorithm for "resampled direct simulation" is a generalisation of the particle filter of Fearnhead (2004) which allows for merging identical/similar particles prior to resampling. We compare the proposed algorithm with this particle filter and with the Gibbs sampler using simulated data and real data sets.

**Keywords** *Direct Simulation, Gibbs Sampler, Importance Sampling, Particle Filters, Perfect Simulation, Rejection Sampling*

## 1 Introduction

Mixture models are commonly used for both density estimation and classification problems (see Titterington *et al.*, 1985; McLachlan and Batsford, 1988, for further details). Bayesian analysis of mixture models has received much interest over the last decade as a result of advances in computational statistics and especially Markov chain Monte Carlo (MCMC) methods (Gilks *et al.*, 1996; Richardson and Green, 1997).

One problem with MCMC methods, both in general and for the specific case of mixture models, is that it can be difficult to diagnose convergence of the MCMC algorithm, and hence difficult to quantify uncertainty in any approximation to the posterior distribution based on the MCMC output. For example, the analysis of the coal-mining disaster data shown in Green (1995) was incorrect due to the MCMC algorithm not converging (compare the results in Green, 2003); and the MCMC analysis of an epidemic SIR

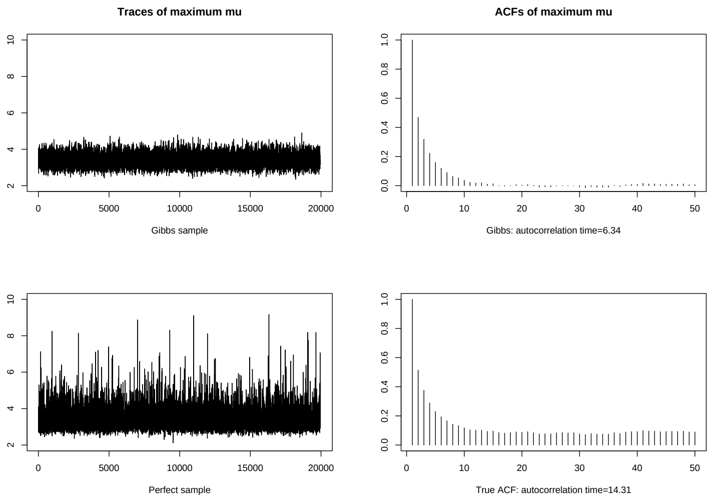

Figure 1: Trace plots and acf plots for the larger of the two Normal means. The top plots summarise the output from the Gibbs sampler while the bottom trace is based on independent draws from the true posterior distribution and the bottom acf plot of the Gibbs output is based on the mean of the true posterior distribution.

model in O'Neill and Roberts (1999) also appears to be inaccurate (see Fearnhead and Meligkotsidou, 2004). Both theoretical and empirical evidence of problems that MCMC algorithms can have in exploring the tails of a posterior distribution are given in Roberts (2003). While Celeux *et al.* (2000) argue that "almost the entirety of MCMC samplers implemented for mixture models has failed to converge".

To demonstrate these difficulties we simulated 50 data points from a 2-component normal mixture model (with common variance), and performed a Bayesian analysis of the data using a Gibbs sampler. Summary of the output of the Gibbs sampler is given in Figure 1 where we show a trace plot and autocorrelation function (acf) plot for the larger of the two normal means. For comparison we also show a trace plot based on independent draws from the true posterior distribution (obtained by a generalisation of the method of Fearnhead, 2005, see Section 3) and an acf plot of the Gibbs output based on the mean of the true posterior distribution of the larger normal mean. It can be seen that the Gibbs sampler misses the upper tail of the posterior distribution, and this is hard to diagnose from the acf plot.

Problems with diagnosing convergence of MCMC algorithms have motivated work in the area of *perfect simulation* - methods based on the "coupling from the past" idea of Propp and Wilson (1996) that enables MCMC algorithms to be constructed so that they produce draws from the true posterior distribution. Attempts have been made to apply these ideas to mixture models (Hobert *et al.*, 1999; Casella *et al.*, 2002) but the resulting algorithms have either limited applicability or are impracticable for even simple real-life problems. More recently Fearnhead (2005) has described a *direct simulation* method that enables independent draws from the true posterior to be made.

Whilst more practicable than perfect simulation, the direct simulation method in Fearnhead (2005) can only be applied to discrete data, relatively small data sets (of the order of 100 to 1000 data points) and small numbers of components (2 or 3). In this paper we describe extensions of the direct simulation method that allows it to be applied to continuous mixture models and data sets of larger size, but again for the analysis of models with a small number of components (i.e up to 3 components).

The outline of the paper is as follows. In the next section we introduce the class of mixture models we will be considering and briefly describe the direct simulation method for discrete data. In Section 3 we extend direct simulation to continuous data. The basic idea is to discretise the data, apply the direct simulation method to the discretised observations, and then correct for this approximation using rejection sampling or importance sampling. In Section 4 we describe the relationship between direct simulation and particle filters, and show how resampling ideas used for particle filters can be applied to reduce the computational cost of direct simulation. In Section 5 we perform comparisons of our method with the Gibbs sampler based on simulated data sets, and in Section 6 we compare our method with the particle filter of Fearnhead (2004) based on real data. The paper ends with a discussion.

## 2 Direct Simulation for Discrete Mixture

We consider mixture models of the form

$$f(x|\beta) = \sum_{k=1}^{K} p_k f(x|\theta_k), \tag{1}$$

where $\beta = (p_1, \dots, p_K, \theta_1, \dots, \theta_K)$ is the set of parameters, $K$ is known, and here and throughout we use the generic notation of $f(\cdot)$ for a probability density or mass function.

Consider data $\mathbf{x} = (x_1, \dots, x_n)$, and let $\mathbf{z} = (z_1, \dots, z_n)$ where $z_k$ denotes the component of (1) that the observation $x_k$ is drawn from. We will assume Dirichlet priors on $\mathbf{p} = (p_1, \dots, p_K)$ and independent conjugate priors on $\theta = (\theta_1, \dots, \theta_K)$ so that the conditional distribution $f(\beta|\mathbf{x},\mathbf{z})$ can be calculated analytically. We have that $f(\mathbf{p}|\mathbf{x},\mathbf{z})$ is just Dirichlet and we have independent posteriors for $f(\theta_k|\mathbf{x},\mathbf{z})$ for $k = 1,\ldots,K$. The Dirichlet distribution on $\mathbf{p}$ depends on $(\mathbf{x},\mathbf{z})$ solely through the numbers of observations allocated to each component. We further assume that $f(\theta_k|\mathbf{x},\mathbf{z})$ depends on $(\mathbf{x},\mathbf{z})$ through a finite set of summary statistics. We denote by $\mathbf{s}(\mathbf{x},\mathbf{z})$ the set of summary statistics, which will be the union of the number of allocations of observations to each component and the summary statistics for the $f(\theta_k|\mathbf{x},\mathbf{z})$s, for which

$$f(\beta|\mathbf{x},\mathbf{z}) = f(\beta|\mathbf{s}(\mathbf{x},\mathbf{z})).$$

In the following we write $\mathbf{s}$ for $\mathbf{s}(\mathbf{x},\mathbf{z})$ when the meaning is clear.

**Example 1: Poisson Mixture**

Consider the case where $f(x|\theta_k)$ denotes the probability mass function of a Poisson random variable with mean $\theta_k$. Assume a Dirichlet prior on $\mathbf{p}$ with parameters $\alpha$ and independent gamma priors on $\theta_1,\ldots,\theta_K$ with parameters $\mathbf{a}$ and $\mathbf{b}$. Then conditional on $(\mathbf{x},\mathbf{z})$ we have independent gamma distributions for $\theta_k$ whose parameters depend on $(\mathbf{x},\mathbf{z})$ solely through the number of observations allocated to component $k$, $n_k$, and their sum, $t_k$. Formally

$$\begin{aligned} f(\beta|\mathbf{x},\mathbf{z}) &= f(\beta|\mathbf{s}(\mathbf{x},\mathbf{z})) \\ &= \mathrm{Dir}(\mathbf{p};\alpha+\mathbf{n})\prod_{k=1}^{K}\mathrm{Gam}(\theta_k;a_k+t_k,b_k+n_k), \end{aligned}$$

where $\mathrm{Dir}(\mathbf{x};\alpha)$ denotes the probability density of a Dirichlet distribution with parameters $\alpha$ evaluated at $\mathbf{x}$, and $\mathrm{Gam}(x;a,b)$ denotes the probability density of a gamma distribution with parameters $a,b$ evaluated at $x$. (For full details of these calculations see Fearnhead, 2005).

Both for this specific example, and for general models which satisfy our assumptions, we can write

$$f(\beta|\mathbf{x}) = \sum_{\mathbf{z}} f(\mathbf{z}|\mathbf{x})f(\beta|\mathbf{x},\mathbf{z}), \tag{2}$$

where

$$f(\mathbf{z}|\mathbf{x}) \propto \int f(\mathbf{x},\mathbf{z}|\beta)f(\beta)\mathrm{d}\beta = \int f(\mathbf{s}|\beta)f(\beta)\mathrm{d}\beta. \tag{3}$$

We define $f(\mathbf{s})$ to be the right-hand side of this equation. Now (2) is proportional to

$$\sum_{\mathbf{z}} f(\mathbf{s})f(\beta|\mathbf{s}(x,\mathbf{z})) = \sum_{\mathbf{s}} M(\mathbf{s};\mathbf{x})f(\mathbf{s})f(\beta|\mathbf{s}), \tag{4}$$

where $M(\mathbf{s};\mathbf{x})$ is the *multiplicity* of sufficient statistics $\mathbf{s}$, the number of allocations of observations $\mathbf{x}$ to components that produce this specific value of sufficient statistics. This rearrangement is based on drawing together like terms in (2).

The usefulness of (4) is that, for discrete mixture models, while the number of terms in (2) increases exponentially with $n$, the number of terms in the right-hand side of (4) increases only as a polynomial in $n$. Furthermore, there is a filtering recursion that enables the multiplicites to be calculated efficiently. The idea behind the direct simulation method of Fearnhead (2005) is to calculate these multiplicities and then simulate directly from (4).

The recursion for the multiplicities is as follows. Denote by $\mathbf{x}_{1:i} = (x_1, \dots, x_i)$. It is possible to calculate the $M(\mathbf{s};\mathbf{x}_{1:i})$s from the $M(\mathbf{s};\mathbf{x}_{1:i-1})$s. Let $\mathbf{s}_k$ denote the value of the summary statistics for data $\mathbf{x}_{1:i-1}$ which will produce a value $\mathbf{s}$ if the $i$th observation is allocated to component $K$. Then $M(\mathbf{s};\mathbf{x}_{1:i}) = \sum_{k=1}^{K} M(\mathbf{s}_k;\mathbf{x}_{1:i-1})$. In practice some of the $M(\mathbf{s}_k;\mathbf{x}_{1:i-1})$ may be zero, and calculating the multiplicities for summaries of data $\mathbf{x}_{1:i}$ is most easily achieved by (i) calculating new values of the summary statistics for all pairs of possible summaries of $\mathbf{x}_{1:i-1}$ and allocations of $x_i$ to components; and (ii) merging identical values of the summary statistics that are produced. If there are $N$ new values of the summary statistics calculated in (i), then the merging in (ii) can be achieved in $O(N \log N)$ time. For an example see Figure 2 and for fuller details see Fearnhead (2005).

Simulation from $f(\beta|\mathbf{x},\mathbf{z})$ is possible by (i) calculating the normalising constant of the right-hand side of (4); (ii) simulating $\mathbf{s}$ from a discrete distribution which has probabilities proportional to $M(\mathbf{s};\mathbf{x})f(\mathbf{s})$; and (iii) simulating from $f(\beta|\mathbf{s})$. Simulation of samples of arbitrary size is possible efficiently (see Fearnhead, 2005). The normalising constant calculated in (i) is the evidence of the model, and can be used to calculate Bayes Factors for comparing different models.

Furthermore, it is possible to simulate from $f(\mathbf{z}|\mathbf{x})$ by replacing (iii) above by (iii') simulate from $f(\mathbf{z}|\mathbf{s})$. The distribution $f(\mathbf{z}|\mathbf{s})$ is uniform over all allocations which produce the required value of the sufficient statistic and it is possible to simulate from this distribution using a backward simulation method (see Appendix A for details).

This direct simulation method has polynomial storage and computational cost. The order of the polynomial depends on the dimension of the smallest set of summary statistics. For the Poisson mixture example the dimension of the set of summary statis-

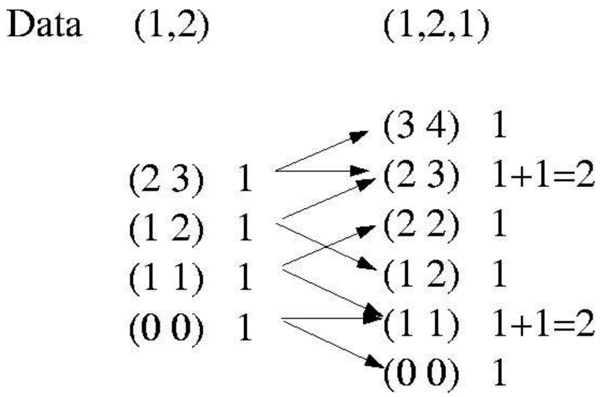

Figure 2: The possible values of the summary statistics and their multiplicities for the data set $1, 2, 1$ for a 2-component Poisson model. The summary statistics, shown in brackets, are $(n_1, t_1)$, the number and sum of the observations allocated to the first component - this uniquely determines the set of summary statistics as the total number and sum of the observations is known. The arrows show the possible values of the summary statistics for this data set for each of the summary statistics based on data (1,2); each of the latter summary statistics has 2 arrows for the possible allocations of the third observation to either the first or second component. The multiplicity of a set of summary statistics for the complete data is obtained by summing the multiplicities of all summary statistics for (1,2) that lead to it.

tics is $2(K-1)$, as it is necessary to store both the number of observations allocated to components $1,\ldots,K-1$ and their sum (the number and sum of the observations allocated to component $K$ is uniquely defined by these summary statistics for the other components). The computational cost for the Poisson example is thus a polynomial of order $2(K-1)$; and direct simulation is possible for data sets of up to around 1,000 for $K=2$ and 100 for $K=3$.

## 3 Direct Simulation for Continuous Mixture

The reason that direct simulation is possible for discrete mixture models is that there can be many allocations of observations to components which produce the same values of the summary statistics. For continuous data this is not the case: each allocation of observations to components will produce a distinct value of the summary statistics, though many of these values will be very close. To adapt the direct simulation approach to continuous data we propose merging terms in (4) that have similar values of the summary statistics. The simplest method of achieving this is to discretise the data $\mathbf{x}$ to $\mathbf{x}^*$ and apply direct simulation to the resulting discrete data. This produces independent samples from the approximate posterior $f(\beta|\mathbf{x}^*)$, and the approximation can be corrected using rejection sampling or importance sampling. (The level of discretisation used to produce $\mathbf{x}^*$ leads to a trade-off between the computational cost and the accuracy of the approximation of $f(\beta|\mathbf{x})$ by $f(\beta|\mathbf{x}^*)$). It is also possible to use importance sampling to approximate the evidence for a given model.

To implement either rejection or importance sampling we work with the distributions $f(\mathbf{z}|\mathbf{x}^*)$ and $f(\mathbf{z}|\mathbf{x})$. Given a sample from $f(\mathbf{z}|\mathbf{x})$ it is straightforward to produce a sample from $f(\beta|\mathbf{x})$, as it is easy to simulate from $f(\beta|\mathbf{x},\mathbf{z})$.

Direct simulation is possible from $f(\mathbf{z}|\mathbf{x}^*)$ and the importance sampling weight is

$$w(\mathbf{z}) \propto \frac{f(\mathbf{z}|\mathbf{x})}{f(\mathbf{z}|\mathbf{x}^*)} \propto \frac{f(\mathbf{s})}{f(\mathbf{s}^*)},$$

where $\mathbf{s}$ and $\mathbf{s}^*$ are the summary statistics for allocation $\mathbf{z}$ and the real and discretised data respectively. (The probabilities $f(\mathbf{s})$ are defined by the right-hand side of equation 3). Importance sampling produces a weighted sample from $f(\mathbf{z}|\mathbf{x})$ by first simulating values from $f(\mathbf{z}|\mathbf{x}^*)$ and assigning each simulated value of $\mathbf{z}$ a weight proportional to $w(\mathbf{z})$.

Rejection sampling requires a bound $L$ to be calculated such that $w(\mathbf{z}) < L$ for all $\mathbf{z}$. If such a bound can be calculated (see below for an example) then independent samples from $f(\mathbf{z}|\mathbf{x})$ are obtained by (i) simulating a $\mathbf{z}'$ from $f(\mathbf{z}|\mathbf{x}^*)$ and (ii) accepting $\mathbf{z}'$ with probability $w(\mathbf{z}')/L$.

**Example 2: Normal Mixture**

Let $f(x|\theta_k)$ denote the probability distribution of a normal random variable with parameters $\theta_k = (\mu_k, \sigma_k^2)$. Assume a Dirichlet prior on $\mathbf{p}$ with parameters $\alpha$, independent inverse gamma priors on the $\sigma_k^2$'s with parameters $a_k$ and $b_k$, and independent normal priors on the $\mu_k$'s of the form $N(\xi_k, \tau_k^2\sigma_k^2)$, $k = 1, \ldots, K$. Then conditional on the observed data $\mathbf{x}$ and the latent allocation variables $\mathbf{z}$ the posterior distribution of the model parameters depends on $(\mathbf{x}, \mathbf{z})$ only through a set of summary statistics $\mathbf{s}(\mathbf{x}, \mathbf{z})$: the number of observations allocated to component $k$, $n_k$, their sum, $t_k$, and the sum of the squared observations, $r_k$, i.e.

$$\begin{aligned} f(\beta|\mathbf{x},\mathbf{z}) &= f(\beta|\mathbf{s}(\mathbf{x},\mathbf{z})) \\ &= \mathrm{Dir}(\mathbf{p}; \alpha + \mathbf{n}) \prod_{k=1}^{K} \mathrm{N}\left(\mu_k; \frac{\xi_k + \tau_k^2 t_k}{1 + n_k\tau_k^2}, \frac{\tau_k^2\sigma_k^2}{1 + n_k\tau_k^2}\right) \\ &\times \prod_{k=1}^{K} \mathrm{IG}\left(\sigma_k^2; a_k + \frac{n_k}{2}, b_k + \frac{1}{2}(r_k - \frac{t_k^2}{n_k}) + \frac{n_k(\xi_k - t_k/n_k)^2}{2(1 + n_k\tau_k^2)}\right), \end{aligned}$$

where $\mathrm{IG}(x; a, b)$ denotes the probability density of an inverse gamma distribution with parameters $a, b$ evaluated at $x$, and $\mathrm{N}(x; \mu, \sigma^2)$ denotes the probability density of a normal distribution with parameters $\mu, \sigma^2$ evaluated at $x$.

The posterior distribution $f(\beta|\mathbf{x})$ can be written in the form of (2), while, if we consider the discretised data $\mathbf{x}^*$, then the approximate posterior $f(\beta|\mathbf{x}^*)$ can be written in the form of (4). Direct simulation is possible from $f(\mathbf{z}|\mathbf{x}^*)$ which is used as an importance function in order to sample from $f(\mathbf{z}|\mathbf{x})$. The importance sampling weight is given by

$$w(\mathbf{z}) \propto \prod_{k=1}^{K} \left[\frac{2b + r_k^* - t_k^{*2}/n_k + n_k(1 + n_k\tau_k^2)^{-1}(\xi_k - t_k^*/n_k)^2}{2b + r_k - t_k^2/n_k + n_k(1 + n_k\tau_k^2)^{-1}(\xi_k - t_k/n_k)^2}\right]^{a_k + n_k/2}.$$

A weighted sample from $f(\beta|\mathbf{x})$ is obtained by first obtaining a weighted sample of allocations $\mathbf{z}$ from $f(\mathbf{z}|\mathbf{x})$, with weights proportional to $w(\mathbf{z})$, and then simulating the corresponding parameter values from $f(\beta|\mathbf{x}, \mathbf{z})$.

The form of $f(\beta|\mathbf{x}, \mathbf{z})$ and the expression for the importance sampling weights is similar for the case of a normal mixture with different component specific means $\mu_k$, $k =$

| | Data × | RS acceptance | ESS | CPU time |
|---|---|---|---|---|
| (i) | 1 | 0.0004 | 5034 | 0.22 |
| (ii) | 5 | 0.0264 | 7819 | 0.55 |
| (iii) | 10 | 0.1435 | 9393 | 1.36 |
| (iv) | 20 | 0.3780 | 9798 | 3.53 |

Table 1: Average acceptance probabilities of the rejection sampling, average effective sample sizes of 10,000 draws from the importance sampling, and average CPU times for the calculation of the approximate posterior based on the discrete (i) original data, (ii) data multiplied by 5, (iii) data multiplied by 10 and (iv) data multiplied by 20.

$1, \ldots, K$, but common variance $\sigma^2$ (see Casella *et al.* (2002) for the two-component mixture). In this case there are two summary statistics, $n_k$ and $t_k$. For the special case that $K = 2$ a bound $L$ can be calculated such that $w(\mathbf{z}) < L$ for all $\mathbf{z}$ (see Appendix B for details) and, therefore, independent samples can be simulated from $f(\mathbf{z}|\mathbf{x})$ using rejection sampling, leading to an exact sample from $f(\beta|\mathbf{x})$.

We have simulated 100 data sets from a 2-component normal mixture with common variance, each consisting of 50 observations. We have analysed these data sets following the approach detailed above, for different levels of discretisation of the data, using both importance sampling and rejection sampling. The results are shown in Table 1. We report the average acceptance probability of the rejection sampling, the average effective sample size (ESS) of Liu (1996) based on 10,000 draws from the importance sampling scheme, and the average CPU time for the calculation of the approximate posterior based on the discrete (i) original data, (ii) data multiplied by 5, (iii) data multiplied by 10 and (iv) data multiplied by 20.

It can be seen that the importance sampling approach works well in terms of both efficiency and accuracy; the variance of the importance weights is small enough that we can have confidence in the results. Furthermore, the sample obtained by this importance sampling scheme is an accurate approximation to the true posterior (see Figure 3 for a comparison with the perfect sample obtained via rejection sampling). For the analysis of a sample of 50 observations from a two-component normal mixture with different variances the average effective sample size of 10,000 draws from the importance sampling scheme is 3576 and the average CPU time is 8.52 sec. Rejection sampling is efficient only for a low level of discretisation of the data. Furthermore, given that it is difficult

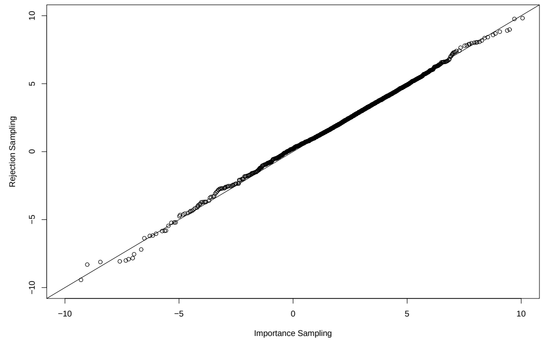

Figure 3: QQ plot for the comparison of a perfect sample from the posterior distribution of the mean of a two-component normal mixture with common variance obtained via rejection sampling with a sample obtained via importance sampling.

to calculate a bound for $K > 2$ or for more complicated mixture models, like the normal mixture with different variances, we do not advocate the use of rejection sampling in general.

## 4 Use of Resampling

The recursions used to calculate the multiplicities, and hence the probabilities of each term in (4) are related to particle filters - sequential Monte Carlo methods for analysing time series data (see Liu and Chen, 1998; Doucet *et al.*, 2001, for an introduction). A key idea within particle filters is to use resampling, and we now describe how resampling ideas can be used with direct simulation to limit the computational cost and make it possible to analyse large data sets. For simplicity we describe the approach for discrete data (for continuous data, resampling can still be used provided the data is discretised first). We call the resulting approach "resampled direct simulation" (RDS), and note that this only produces an approximate sample from the posterior distribution.

We introduce artificial time so that the data at time $i$ is $\mathbf{x}_{1:i}$. The posterior at time $i$ is

$$f(\beta|\mathbf{x}_{1:i}) \propto \sum_{\mathbf{s}^{(i)}} M(\mathbf{s}^{(i)};\mathbf{x}_{1:i})f(\mathbf{s}^{(i)})f(\beta|\mathbf{s}^{(i)}). \quad (5)$$

To relate this to particle filters, we can think of the values of $\mathbf{s}^{(i)}$ as particles and $q(\mathbf{s}^{(i)}) = M(\mathbf{s}^{(i)};\mathbf{x}_{1:i})f(\mathbf{s}^{(i)})$ as the weight assigned to particle $\mathbf{s}^{(i)}$.

Particles and their weights at time $i$ can be calculated recursively from the weighted particles at time $i-1$. Specifically if $\mathbf{s}_k^{(i-1)}$ denotes the particle at time $i-1$ which would produce particle $\mathbf{s}^{(i)}$ at time $i$ if observation $x_i$ is allocated to component $k$, then

$$q(\mathbf{s}^{(i)}) = \sum_{k=1}^{K} q(\mathbf{s}_k^{(i-1)}) \frac{f(\mathbf{s}^{(i)})}{f(\mathbf{s}_k^{(i-1)})}.$$

Note that in practice some of the $\mathbf{s}_k^{(i-1)}$ will have zero weight, and hence would not be included in this sum. In practice, we would calculate the set of weighted particles at time $i$ by first propagating all particles at time $i-1$ and then merging identical particles at time $i$.

Resampling can be used within such a particle filter to limit the computational cost (see Chen and Liu, 2000; Fearnhead and Clifford, 2003). Assume we wish to store at most $N$ particles at any time. If we currently have more than $N$ particles, we resample $N$ particles, update their weights, and then approximate $f(\beta|\mathbf{x}_{1:i})$ based on these $N$ new weighted particles. Various resampling algorithms have been suggested for particle filters (Kitagawa, 1996; Carpenter *et al.*, 1999; Liu *et al.*, 1998), but we use the scheme of Fearnhead and Clifford (2003) which is optimal, in terms of minimising a square error condition on the new weights, over all unbiased resampling schemes.

This resampling scheme proceeds as follows:

(i) Assume that our approximation to $f(\beta|\mathbf{x}_{1:i})$ is based on $L$ particles $\mathbf{s}_1^{(i)}, \ldots, \mathbf{s}_L^{(i)}$ with weights $q_1^{(i)}, \ldots, q_L^{(i)}$.

(ii) Calculate $c$ the solution of $N = \sum_{l=1}^{L} \min(cq_l^{(i)}, 1)$.

(iii) For $l = 1, \ldots, L$, if $cq_l^{(i)} > 1$ then keep particle $l$ with the same weight. Assume $N^*$ such particles are kept.

(iv) Use the stratified sampling algorithm of Carpenter *et al.* (1999) to simulate $N - N^*$ particles from the remaining $L - N^*$ particles. Assign each resampled particle a weight $1/c$.

The rationale for this algorithm is given in Fearnhead and Clifford (2003).

Simulation from the final approximation of $f(\beta|\mathbf{x})$ is straightforward. To simulate from the approximation of $f(\mathbf{z}|\mathbf{x})$ is possible as described in Appendix C.

**Example 1 Poisson Mixture (revisited)**

We applied this resampling algorithm to a 3-component Poisson mixture model. We analysed data from Leroux and Puterman (1992) of the number of movements by a fetal lamb over 240 consecutive 5 second periods (see Table 2).

| Movements | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Frequency | 182 | 41 | 12 | 2 | 2 | 0 | 0 | 1 |

Table 2: Number of movements by a fetal lamb over 240 consecutive 5 second periods.

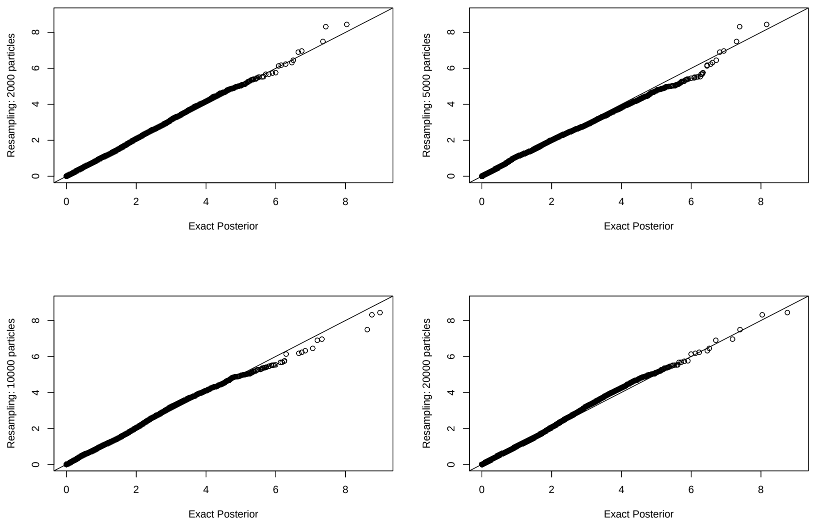

Figure 4: Quantile-quantile plots for the comparison of a perfect sample from the posterior with samples obtained using RDS for different numbers of particles stored.

We compared the approximate sample we obtained using RDS to a perfect sample from direct simulation. A comparison of the samples for different values of $N$ is given in Figure 4. It can be seen that the approximate samples obtained by RDS are quite accurate even for moderate values of $N$.

## 5 Comparison with MCMC

In this section we compare the RDS approach with the Gibbs sampler. The comparison of these two algorithms is based on analyses of simulated data and it is made in terms of the computational efficiency of the algorithms.

We consider different simulated data sets from 2-component normal mixtures and summarise the performance of each method on a given data set using the effective sample size (ESS) of Carpenter *et al.* (1999). This is done by running both algorithms independently 100 times on each data set. For a specific function of the output, the ESS is given by the ratio of the estimate of the posterior variance of the function to the variance of the posterior mean of the function across the independent runs of the algorithm. This measure of efficiency can be used for both particle filters and MCMC algorithms. If an algorithm has an ESS of $E$, then inference based on this algorithm is roughly as accurate as inference based on $E$ independent draws from the full posterior distribution. (Note that this ESS is different to that of Liu, 1996, used in Section 3 - though the interpretation of the value is the same).

A comparison based on the ESS is equivalent to a comparison based on the variability of estimates of a given function produced by the different algorithms considered, and is sensible here as both algorithms produce similar esimates of the posterior means of the functions we consider.

The results we present are based on using the same number of particles for RDS as the number of iterations of the Gibbs sampler (after burn-in). The particles in RDS produce a discrete distribution on the allocation of observations to components. We produce a sample from the posterior distribution of the parameters by resampling independently from the distribution of allocations, and then sample a value of the parameters conditional on the allocations. (Actually, we produce a weighted sample due to the need for importance sampling weights for each simulated allocation). As simulating from the distribution of allocations is much quicker than producing the set of particles, we resample a larger number of parameter values than the number of particles.

A fair comparison of the two algorithms would take into account the different CPU costs of the algorithms. Direct comparisons of the CPU costs is difficult due to variations in the efficiency of the computer code used, and the speed of the computing language. However, rough comparisons can be made which will aid the interpretation of the results we present. For our experiments, the CPU cost of running the RDS algorithm with $N$ particles is about 2 times the cost of $N$ iterations of the Gibbs sampler. The RDS has an extra cost due to resampling parameter values from the final set of particles, and a further computational cost due to the merging of particles prior to resampling. The Gibbs sampler has the extra cost for the burn-in period, which is relatively small.

We have considered data sets of size $n = 50$, 100 from 3 different 2-component mixtures. In each case the mixture components were labelled according to an identifiability constraint on the component means; the smallest mean was associated with the first component. The mixing proportion of the first component was 0.6, and the component specific parameters were chosen to have different degrees of overlap between the two components. (i) $\mu_1 = 1$, $\mu_2 = 2$, $\sigma_1^2 = 1.5$, $\sigma_2^2 = 2$, (ii) $\mu_1 = 1$, $\mu_2 = 3$, $\sigma_1^2 = 1$, $\sigma_2^2 = 2$, and (ii) $\mu_1 = 1$, $\mu_2 = 5$, $\sigma_1^2 = 1$, $\sigma_2^2 = 2$. All data sets were analysed under the prior specification of Section 3, with $\alpha = (1, 1)$, $a_1 = a_2 = 1$, $b_1 = b_2 = 1$, $\xi_1 = \xi_2 = 2$ or $\xi_1 = \xi_2 = 3$ and $\tau_1^2 = \tau_2^2 = 1$.

We analysed these data sets using (a) 2000 particles for the RDS with 5000 resampled draws, and 2500 Gibbs iterations with the first 500 draws being discarded, and (b) 5000 particles for the RDS with 10000 resampled draws, and 6000 Gibbs iterations with the first 1000 draws being discarded. For the RDS we discretised the simulated observations to the nearest integer. The ESS values for some of the model parameters as well as for some tail posterior probability for the smallest of the two means are shown in Tables 3-5.

It can be seen that the RDS performs much better than the Gibbs sampler in all cases. The efficiency of both algorithms decreases with the number of observations and, in general, it is greater at estimating the component variances than at estimating the component means. The RDS can be up to 20 times more efficient than the Gibbs sampler. However, taking into account the fact that the CPU cost for the RDS is roughly 2 times that of the Gibbs sampler, we conclude that the RDS can be up to one order of magnitude more efficient than the Gibbs, at analysing normal mixture models.

Another advantage of the RDS over the Gibbs sampler is that it enables us to estimate the evidence of the model. Our simulation studies have shown that the evidence is actually estimated very accurately. In all our examples, the standard error of the estimate of the logarithm of the evidence over 100 replications was less than 0.005.

## 6 Comparison with Particle Filter

As discussed in Section 4, the RDS algorithm is actually a particle filter. Particle filters for analysing mixture models do exist (Ishwaran *et al.*, 2001; Fearnhead, 2004). The general approach is that, after having processed sequentially the first $i$ observations, each of the $N$ current particles consists of a specific assignment of the $i$ observations to components. The $N$ particles are propagated as each new observation is processed by considering for each particle all the possible assignments of the $(i + 1)$st observation to a component. Then, $N$ new particles are produced for the first $(i + 1)$ observations by resampling the possible particles.

|  | 50 observations |  |  |  | 100 observations |  |  |  |
|---|---|---|---|---|---|---|---|---|
|  | Gibbs |  | RDS |  | Gibbs |  | RDS |  |
|  | 2000 | 5000 | 2000 | 5000 | 2000 | 5000 | 2000 | 5000 |
|  | draws | draws | particles | particles | draws | draws | particles | particles |
| $\mu_1$ | 375 | 533 | 3165 | 4572 | 215 | 306 | 1409 | 1891 |
| $\sigma_1^2$ | 1935 | 5141 | 5115 | 10021 | 1948 | 5176 | 4186 | 8200 |
| $p_1$ | 303 | 518 | 3217 | 3728 | 214 | 366 | 1441 | 1670 |
| $\Pr(\mu_1 < 0)$ | 401 | 649 | 2892 | 5776 | 229 | 371 | 1676 | 3348 |

Table 3: Comparison of the efficiency of RDS and the Gibbs sampler at analysing 50 and 100 observations from the normal 2-component mixture with parameters $\mu_1 = 1$, $\mu_2 = 2$, $\sigma_1^2 = 1.5$, $\sigma_2^2 = 2$ and $p_1 = 0.6$.

|  | 50 observations |  |  |  | 100 observations |  |  |  |
|---|---|---|---|---|---|---|---|---|
|  | Gibbs |  | RDS |  | Gibbs |  | RDS |  |
|  | 2000 | 5000 | 2000 | 5000 | 2000 | 5000 | 2000 | 5000 |
|  | draws | draws | particles | particles | draws | draws | particles | particles |
| $\mu_1$ | 428 | 982 | 1922 | 4405 | 280 | 643 | 564 | 1292 |
| $\sigma_1^2$ | 2023 | 5239 | 3662 | 7439 | 1686 | 4366 | 6623 | 13454 |
| $p_1$ | 117 | 333 | 2307 | 8449 | 93 | 207 | 514 | 1149 |
| $\Pr(\mu_1 < 0)$ | 881 | 2688 | 3189 | 6458 | 232 | 707 | 1036 | 2098 |

Table 4: Comparison of the efficiency of RDS and the Gibbs sampler at analysing 50 and 100 observations from the normal 2-component mixture with parameters $\mu_1 = 1$, $\mu_2 = 3$, $\sigma_1^2 = 1$, $\sigma_2^2 = 2$ and $p_1 = 0.6$.

| | 50 observations | | | | 100 observations | | | |
|---|---|---|---|---|---|---|---|---|
| | Gibbs | | RDS | | Gibbs | | RDS | |
| | 2000 | 5000 | 2000 | 5000 | 2000 | 5000 | 2000 | 5000 |
| | draws | draws | particles | particles | draws | draws | particles | particles |
| $\mu_1$ | 592 | 1437 | 2205 | 3816 | 78 | 162 | 356 | 562 |
| $\sigma_1^2$ | 394 | 1187 | 3071 | 6346 | 77 | 203 | 581 | 1285 |
| $p_1$ | 156 | 299 | 3996 | 11928 | 88 | 170 | 657 | 1812 |
| $\Pr(\mu_1 < 0)$ | 1523 | 3982 | 5170 | 16216 | 996 | 2604 | 1673 | 5205 |

Table 5: Comparison of the efficiency of RDS and the Gibbs sampler at analysing 50 and 100 observations from the normal 2-component mixture with parameters $\mu_1 = 1$, $\mu_2 = 5$, $\sigma_1^2 = 1$, $\sigma_2^2 = 2$ and $p_1 = 0.6$.

The particle filters of Ishwaran *et al.* (2001) and Fearnhead (2004) were designed for analysing a mixture model where the number of components is unknown; however they can be easily applied to the case of a known number of components. For such a case, the particle filter (PF) of Fearnhead (2004) is closely related to our RDS algorithm. For discrete data, the difference between the PF and RDS is that in RDS particles with identical values of the sufficient statistics are merged prior to resampling. For continuous data, the PF can be viewed as a special case of RDS, as for such data the amount of merging of particles depends on the level of discretisation used. As we discretise the data less, then there will be less merging of particles, and the PF is the version of RDS we obtain with no discretisation, and hence no merging of particles.

In this section we compare the RDS algorithm with the PF of Fearnhead (2004) in terms of the computational efficiency of the two algorithms. Our comparison is based on analysing two real data sets using 2-component and 3-component Poisson mixtures. The first data set concerns the number of death notices of women, 80 years of age and older, which appeared in the London Times each day for a year period (Schilling, 1947). This data is given in Table 6. The second data set consists of the fetal lamb movements data (Leroux and Puterman, 1992) of Table 2.

We used the ESS measure of efficiency (Carpenter *et al.*, 1999) to compare the performance of the two algorithms, computed after running both algorithms independently 100 times on each data set. The results we present are based on using the same number of particles for RDS and for the PF. We resample a much larger number of parameter

| Death Count | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Frequency | 162 | 267 | 271 | 185 | 111 | 61 | 27 | 8 | 3 | 1 |

Table 6: Number of death notices of women, 80 years and older, in the London Times each day for a year period.

| | 2-component mixture | | | | 3-component mixture | |
|---|---|---|---|---|---|---|
| | PF | | RDS | | PF | RDS |
| | 2000 particles | 5000 particles | 2000 particles | 5000 particles | 20000 particles | |
| $\theta_1$ | 675 | 1931 | 8612 | 17704 | 2831 | 1787 |
| $p_1$ | 603 | 1882 | 5397 | 18871 | 2289 | 1136 |

Table 7: Comparison of the efficiency of RDS and the PF at analysing the fetal lamb movements data under Poisson mixtures.

values than the number of particles since simulating from the posterior of the parameters is very fast. Again the mixture components were labelled according to an identifiability constraint on the component means. The prior specification was as in Section 2 with $\alpha_k = 1$, $a_k = 1$ and $b_k = 1$, $k = 1, 2$ or $k = 1, 2, 3$.

We analysed the fetal lamb movements data under a 2-component Poisson mixture using (a) 2000 particles and 20000 resampled draws, and (b) 5000 particles and 20000 resampled draws. We also analysed this data under a 3-component Poisson mixture using 20000 particles and 50000 resampled draws. The ESS values for the parameters associated with the first component are shown in Table 7. For comparison the ESS values obtained by analysing the fetal lamb movements data under 2-component and 3-component Poisson mixtures via the Gibbs sampler, with equal number of draws as the number of particles in the particle filters, were between one and two orders of magnitude smaller than those of the RDS algorithm. Finally, we analysed the daily death notices data under a 2-component Poisson mixture using 2000 particles and 20000 resampled draws. The ESS values for $\theta_1$ and $p_1$ for the RDS algorithm were 34 and 28, respectively, while for the PF they were 44 and 26, respectively.

It can be seen that, while the RDS algorithm performs much better at analysing the fetal lamb movements data set under a 2-component mixture, the PF performs better at analysing this data under a 3-component mixture and slightly better at analysing

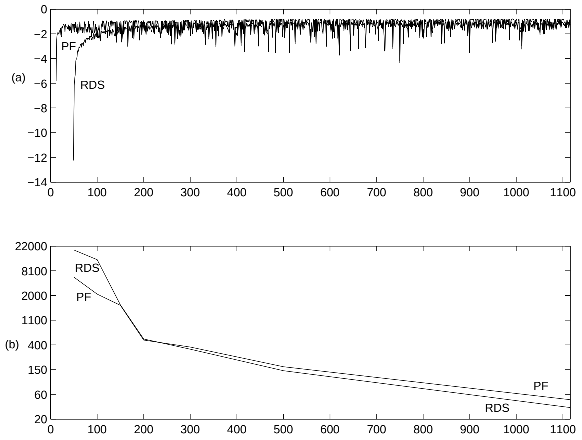

Figure 5: (a) Logarithm of the resampling error against iteration and (b) ESS against the number of observations for the daily death notices data analysed by the RDS algorithm and the particle filter under a 2-component Poisson mixture using 2000 particles.

the large daily death notices data under a 2-component mixture. The reason for this striking difference in the relative accuracy of the two approaches for different data sets can be understood by looking at the resampling error introduced by the RDS and PF algorithms.

In Figure 5 is shown a plot of the resampling error against the iteration of the algorithm for the daily death notices data analysed by the RDS algorithm and the particle filter under a 2-component Poisson mixture using 2000 particles and 20000 resampled draws. For this data set the ESS values obtained by analysing subsets of the data of increasing size were calculated and they are also shown in Figure 5. It can be seen that the resampling error of the RDS is substantially smaller than that of the PF for the first 150 observations, and thereafter each algorithm has a similar amount of error introduced via resampling at each iteration. The plot of the ESS shows that RDS substantially outperforms the PF when analysing the first 150 observations (or fewer), the region for which the resampling error is substantially smaller for RDS; but the two algorithms perform similarly when analysing 200 observations or more, when the resampling errors of the two algorithms are similar.

In Figure 6 are shown plots of the resampling error against the iteration of the algorithm for the fetal lamb movements data analysed by the RDS algorithm and the particle filter under (a) a 2-component Poisson mixture using 5000 particles and 20000 resampled draws and (b) a 3-component Poisson mixture using 20000 particles and 50000 resam-

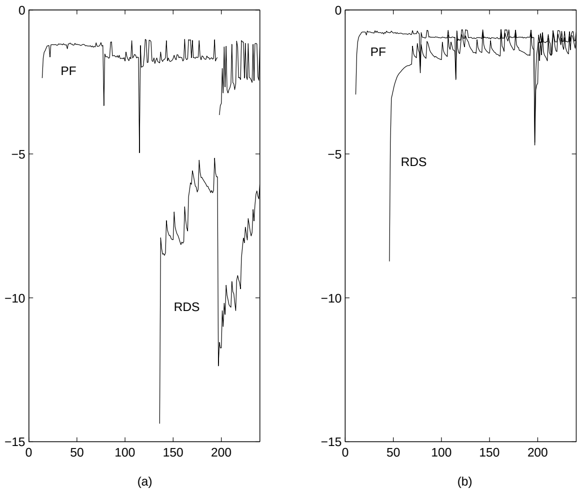

Figure 6: Logarithm of the resampling error against iteration for the fetal lamb movements data analysed by the RDS algorithm and the particle filter under (a) a 2-component Poisson mixture using 5000 particles and (b) a 3-component Poisson mixture using 20000 particles.

pled draws. In the case of the 2-component mixture the resampling error of the RDS is substantially smaller than that of the PF; and hence the substantially better performance in this case. For the 3-component case, the resampling error is similar for the two algorithms for the last 150 observations analysed, and hence the similar performance of the two algorithms.

Thus we see that there is a threshold, in terms of the number of observations, which governs the relative performance of RDS and the PF. If fewer than this number of observations is analysed then RDS will subsantially outperform the PF. Otherwise they will give similar results. This threshold increases with the number of particles. For example the number of observations of the fetal lamb data that can be analysed under a 2-component mixture before a resampling error of greater than 0.001 are 90, 202 and 240 (the whole data set) for 1000, 5000 and 10000 particles respectively.

Table 7 suggests that the PF can even outperform RDS if sufficient observations are analysed, presumably because of the different ways the resampling error accumulates in the two algorithms. When computational time is taken into account (the PF is quicker to implement than RDS, as it does not involve a merging step), RDS should only be preferred to the PF if fewer observations than this threshold are being analysed.

We can use the fact that RDS can often accurately analyse a sub-set of a large data set,

|  | 2-component mixture | 3-component mixture |
|---|---|---|
| $\theta_1$ | 17583 | 7050 |
| $p_1$ | 15253 | 14838 |

Table 8: Efficiency of the importance sampling approach based on RDS at analysing the fetal lamb movement data under a 2-component Poisson mixture (half of the observations were analysed using RDS with 2000 particles) and under a 3-component Poisson mixture (a third of the observations were analysed using RDS with 20000 particles).

to create a simple importance sampling approach for analysing the complete data. The idea is to produce a stratified subset of the complete data set. We stratify the subset in such a way that the proportion of each data value (or range of data values) in the subset is as close as possible to its proportion in the full data-set. We then use RDS to approximate the posterior distribution of the parameters for the sub-set of the data. Finally we use this distribution as a proposal distribution for the full data set. The importance sampling weights are just the ratio of the posterior density of the full data to the posterior density of the subset; which is the likelihood of the extra observations in the full data set.

We have used this approach to analyse the fetal lamb movements data under a 2-component and under a 3-component Poisson mixture. In the case of the 2-component mixture we analysed half of the observations using RDS with 2000 particles and 20000 resampled draws. In the case of the 3-component mixture we analysed a third of the observations using RDS with 20000 particles and 50000 resampled draws. The ESS values for the parameters associated with the first component are shown in Table 8.

## 7 Discussion

In this paper we have extended the direct simulation method of Fearnhead (2005), which enables exact Bayesian inference for discrete mixture distributions to be made, to the case of continuous mixture distributions. The posterior distribution for mixture models depends on the allocation of observations to components only through a set of summary statistics. The direct simulation method is based on a set of forward-backward recursions, which can be used to calculate the exact posterior distribution of the allocations and to simulate samples from it. Our approach was based on discretising the continuous data, applying direct simulation to the discretised observations, and then correcting for this approximation via rejection sampling or importance sampling.

A simple extension of the idea we presented is to discretise the values of the summary statistics for each observation rather than the observation itself. For example, for the normal mixture models we considered, an observation $x_i = 10.4$, say, could be discretised to 10 (the nearest integer), with $x_i^2$ discretised to 110 (the nearest ten).

Rejection sampling, if applicable, is in general preferable to importance sampling, since it provides an exact sample from the posterior distribution. However, calculation of the upper bound for the weights is not possible for complicated mixture models and, even when it is possible, rejection sampling is not efficient for high levels of discretisation of the data. We have shown that a sample from the posterior distribution obtained via importance sampling is almost as accurate as an exact sample obtained via rejection sampling.

Both the original direct simulation method for analysing discrete mixture models and our extension to the case of continuous mixtures become infeasible if the sample size is large. In order to deal with larger data sets we have introduced resampling steps similar to those in particle filters within the steps of the recursion for the calculation of the posterior distribution. The resampled direct simulation is computationally efficient and particularly accurate. However, while resampling enables large data sets to be analysed, it does not allow models with a large (or unknown) number of components to be analysed efficiently.

A comparative study based on simulated data from different 2-component normal mixtures has shown that the RDS can be up to one order of magnitude more efficient than the Gibbs sampler. The advantage of RDS over the Gibbs is greater for Poisson mixture models where no discretisation, and therefore no IS correction, is required. Analysis of subsets of observations from two real data sets under Poisson mixtures suggests that the RDS can be up to 2 orders of magnitude more efficient than the Gibbs at analysing 2-component mixtures and up to 1 order of magnitude more efficient than the Gibbs at analysing 3-component mixtures.

The RDS algorithm is a generalisation of the particle filter of Fearnhead (2004) for mixture models with known number of components. The efficiency of the RDS algorithm is that, even allowing for resampling, many identical particles are produced when the particles are propagated and these particles can be merged. Analysis of two real data sets under Poisson mixtures has shown that there is a threshold number of observations for which the RDS algorithm can be up to an order of magnitude more efficient than the particle filter described in Fearnhead (2004). As the size of the data set increases, the two algorithms tend to be similar.

## Appendix A

To simulate $z_n$ from the marginal distribution $f(z_n|\mathbf{s}, \mathbf{x})$:

(i) Calculate the value, $\mathbf{s}^{(k)}$, of the summary statistics of observations $\mathbf{x}_{1:n-1}$ that would produce $\mathbf{s}$ if $x_n$ is allocated to component $k$.

(ii) Simulate $z_n$ from the discrete distribution that allocates probability $M(\mathbf{s}^{(k)}; \mathbf{x}_{1:n-1})/M(\mathbf{s}; \mathbf{x})$ to value $k$, for $k = 1, \ldots, K$.

Simulating $z_{n-1}$ conditional on $z_n = k$ is equivalent to simulating from the marginal $f(z_{n-1}|\mathbf{s}^k, \mathbf{x}_{1:n-1})$, and can be performed as above. Simulation of $z_{n-2}, z_{n-3}, \ldots, z_1$ proceeds in a similar manner.

## Appendix B

The importance sampling weights $w(\mathbf{z})$ in the case of a two-component normal mixture with common variance are given by

$$w(\mathbf{z}) \propto \left[\frac{2b + \sum_{k=1}^{2} n_k(1 + n_k\tau_k^2)^{-1}(\xi_k - t_k^*/n_k)^2 + \sum_{i=1}^{N} x_i^2 - \sum_{k=1}^{2} t_k^{*2}/n_k}{2b + \sum_{k=1}^{2} n_k(1 + n_k\tau_k^2)^{-1}(\xi_k - t_k/n_k)^2 + \sum_{i=1}^{N} x_i^2 - \sum_{k=1}^{2} t_k^2/n_k}\right]^{a+n/2}.$$

In order to compute a bound for $w(\mathbf{z})$ we note that for fixed values of the sufficient statistics, $(n_1, n_2, t_1^*, t_2^*)$, the value of the weight depends solely on the difference $d = t_1 - t_1^*$ – as the value of $d$ will determine $t_2 - t_2^*$. The weight depends on $d$ through the power of a quadratic in $d$. It is straightforward to show that maximising the weight will thus be achieved at either the largest or smallest possible value of $d$.

We proceed as follows:

(i) Compute the differences $d_i = x_i - x_i^*$ and order them in ascending order.

(ii) For the $j$th possible set of sufficient statistics $\mathbf{s}_j^*(\mathbf{x}, \mathbf{z}) = ((n_1, n_2), (t_1^*, t_2^*))$

(a) allocate the $n_1$ smaller differences to the first component and the $n_2$ larger differences to the second component and set $t_1^{(1)} = t_1^* + \sum_{i=1}^{n_1} d_i$, $t_2^{(1)} = t_2^* + \sum_{i=n_1+1}^{n} d_i$. Compute $w_j^{(1)}(\mathbf{z})$ for $t_k = t_k^{(1)}$, $k = 1, 2$.

(b) allocate the $n_1$ larger differences to the first component and the $n_2$ smaller differences to the second component and set $t_1^{(2)} = t_1^* + \sum_{i=n_2+1}^{n} d_i$, $t_2^{(2)} = t_2^* + \sum_{i=1}^{n_2} d_i$. Compute $w_j^{(2)}(\mathbf{z})$ for $t_k = t_k^{(2)}$, $k = 1, 2$.

(c) Set $L_j = \max(w_j^{(1)}(\mathbf{z}), w_j^{(2)}(\mathbf{z}))$.

(iii) Calculate the bound $L = \max_j L_j$, where the maximum is over all the possible sets of sufficient statistics.

## Appendix C

Simulation from the approximate distribution of $f(z|\mathbf{s}, \mathbf{x})$ obtained under the resampling scheme, is similar to that described in Appendix A. We just describe the approach to simulate $f(z_n|\mathbf{s}, \mathbf{x})$.

(i) Define $\mathbf{s}_1^{(n-1)}, \ldots, \mathbf{s}_K^{(n-1)}$ to be the value of the summary statistics at time $n-1$ which produce $\mathbf{s}$ at time $n$ if $z_n = 1, \ldots, K$ respectively. Then the weight for $\mathbf{s}$ was calculated by

$$q(\mathbf{s}) = \sum_{k=1}^{K} q(\mathbf{s}_k^{(n-1)}) \frac{f(\mathbf{s})}{f(\mathbf{s}_k^{(n-1)})}.$$

(Note that the weight for some of these statistics at time $(n-1)$ may be 0.)

(ii) Set $z_n = k$ with probability proportional to $q(\mathbf{s}_k^{(n-1)}) f(\mathbf{s}_k^{(n-1)})$.

# References

Carpenter, J., Clifford, P. and Fearnhead, P. (1999). An improved particle filter for non-linear problems. *IEE proceedings-Radar, Sonar and Navigation* **146**, 2–7.

Casella, G., Mengerson, K. L., Robert, C. P. and Titterington, D. M. (2002). Perfect samplers for mixtures of distributions. *Journal of the Royal Statistical Society, Series B* **64**, 777–790.

Celeux, G., Hurn, M. and Robert, C. P. (2000). Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association* **95**, 957–970.

Chen, R. and Liu, J. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society, Series B* **62**, 493–508.

Doucet, A., de Freitas, J. F. G. and Gordon, N. J., eds. (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York.

Fearnhead, P. (2004). Particle filters for mixture models with an unknown number of components. *Statistics and Computing* **14**, 11–21.

Fearnhead, P. (2005). Direct simulation for discrete mixture distributions. *Statistics and Computing* **15**, 125–133.

Fearnhead, P. and Clifford, P. (2003). Online inference for hidden Markov models. *Journal of the Royal Statistical Society, Series B* **65**, 887–899.

Fearnhead, P. and Meligkotsidou, L. (2004). Exact Filtering for Partially-observed Continuous-time Models. *Journal of the Royal Statistical Society, Series B* **66**, 771–789.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (1996). *Markov chain Monte Carlo in practice*. London: Chapman and Hall.

Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82**, 711–732.

Green, P. J. (2003). Trans-dimensional Markov chain Monte Carlo. In: *Highly Structured Stochastic Systems* (eds. P. J. Green, N. L. Hjort and S. Richardson), Oxford University Press.

Hobert, J. P., Robert, C. P. and Titterington, D. M. (1999). On perfect simulation for some mixtures of distributions. *Statistics and Computing* **9**, 287–298.

Ishwaran, H., James, L. F. and Sun, J. (2001). Bayesian model selection in finite mixtures by marginal density decompositions. *Journal of the American Statistical Association* .

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics* **5**, 1–25.

Leroux, B. G. and Puterman, M. L. (1992). Maximum-penalized-likelihood estimation for independent and Markov-dependent mixture models. *Biometrics* **48**, 545–558.

Liu, J. S. (1996). Metropolised independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing* **6**, 113–119.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association.* **93**, 1032–1044.

Liu, J. S., Chen, R. and Wong, W. H. (1998). Rejection control and sequential importance sampling. *Journal of the American Statistical Society* **93**, 1022–1031.

McLachlan, G. J. and Batsford, K. E. (1988). *Mixture Models: Inference and Applications to Clustering*. Decker, New York.

O'Neill, P. D. and Roberts, G. O. (1999). Bayesian inference for partially observed stochastic epidemics. *Journal of the Royal Statistical Society, Series A* **162**, 121–129.

Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms* **9**, 223–252.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, series B* **59**, 731–792.

Roberts, G. O. (2003). Linking theory and practice of MCMC (with discussion). In: *Highly Structured Stochastic Systems.*

Schilling, W. (1947). A frequency distribution represented as the sum of two Poisson distributions. *Journal of the American Statistical Association* **42**, 407–424.

Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. Wiley, New York.