Computational Methods for Complex Stochastic Systems: A Review of Some Alternatives to MCMC

Paul Fearnhead

Department of Mathematics and Statistics
Lancaster University
UK

Abstract

We consider analysis of complex stochastic models based upon partial information. MCMC and reversible jump MCMC are often the methods of choice for such problems, but in some situations they can be difficult to implement; and suffer from problems such as poor mixing, and the difficulty of diagnosing convergence. Here we review three alternatives to MCMC methods: importance sampling, the forward-backward algorithm, and sequential Monte Carlo (SMC). We discuss how to design good proposal densities for importance sampling, show some of the range of models for which the forward-backward algorithm can be applied, and show how resampling ideas from SMC can be used to improve the efficiency of the other two methods. We demonstrate these methods on a range of examples, including estimating the transition density of a diffusion and of a discrete-state continuous-time Markov chain; inferring structure in population genetics; and segmenting genetic divergence data.

Keywords: Diffusions, Forward-Backward Algorithm, Importance Sampling, Missing Data, Particle Filter, Population Genetics

1 Introduction

Many scientific models involve a complex latent structure. They model the latent structure via a stochastic process, and then model how the observations relate to the value of this process. Such models can be viewed as "missing data" models: where it is easy to write down the likelihood if our data was both the observations and the value of the latent process, but the data on the latter is missing. To perform inference for parameters requires averaging over the possible realisation of this missing data; and often this requires the use of modern computational statistical methods.

The models we consider consist of observations $\mathbf{y}$, missing data $\mathbf{x}$, and parameters, θ. We assume that the densities $p(\mathbf{x}|\theta)$, and $p(\mathbf{y}|\mathbf{x},\theta)$ are available in closed form. Our interest will be in inference for either θ or for $\mathbf{x}$. For the former we will wish to calculate the likelihood,

$$p(\mathbf{y}|\theta) = \int p(\mathbf{x}|\theta)p(\mathbf{y}|\mathbf{x},\theta)\mathrm{d}\theta, \tag{1}$$

and for the latter to simulate from the conditional distribution of the missing data,

$$p(\mathbf{x}|\mathbf{y},\theta) \propto p(\mathbf{x}|\theta)p(\mathbf{y}|\mathbf{x},\theta). \tag{2}$$

The normalising constant of (2) is just the likelihood, (1). We are interested in the situation where the integral in (1) is intractable, so we need to resort to Monte Carlo methods to estimate (1), or to sample from (2).

A common approach to analysing such data is to use Markov chain Monte Carlo (MCMC) or reversible jump MCMC methods (Gamerman, 2006; Green, 1995). For example, a prior on θ could be specified, and then we could run an MCMC algorithm which has the posterior $p(\theta, \mathbf{x}|\mathbf{y})$ as its stationary distribution. The output from the MCMC algorithm can then be used to make inference about θ, or $\mathbf{x}$, or both. The popularity of MCMC methods can be seen by the fact that the paper that first introduces the Metropolis-Hastings algorithm (Metropolis et al., 1953) has been cited nearly ten thousand times. Despite this popularity, there can be problems with implementing MCMC methods efficiently on complex problems, as can be seen by the ongoing research into how to design and implement MCMC algorithms (Papaspilopoulos et al., 2003), the use of population-based MCMC ideas (Kou et al., 2006; Stephens et al., 2007) and the need for tailored algorithms for many applications (e.g. Redelings and Suchard, 2005).

In this article we describe alternatives to MCMC methods, with a particular emphasis on the author's own research. We do not attempt to compare these with MCMC methods: our aim is primarily to show that there are non-MCMC approaches to missing data problems that could be considered. However there is evidence that for some applications, non-MCMC methods can be more efficient than MCMC algorithms. Example applications include: some multiple changepoint models (Fearnhead, 2006); time-ordered hidden Markov models (Chopin, 2007); some generalised linear models (Chopin, 2002); estimating recombination rates from population data (Fearnhead and Donnelly, 2001); and mixture models (Del Moral et al., 2006; Fearnhead, 2004). For some applications of the methods described here (see Section 3) it is possible to get independent draws from the posterior, which avoids the problems of mixing and diagnosing convergence of MCMC algorithms. Finally there can also be advantages in terms of implementation when the state in the MCMC algorithm is a particularly complex object. For example, for the models considered in Fearnhead and Donnelly (2001) the state of an MCMC algorithm would be a variable dimensional graph, but an Importance Sampling method, which uses the Markov structure of the graph, needs only store a state which is vector-valued.

The common feature of the models we consider is that the missing data has a Markov structure. Thus $\mathbf{X} = (X_1, \ldots, X_\tau)$, where τ is a stopping time, and we can factorise

$$p(\mathbf{x}|\theta) = p(x_1|\theta) \prod_{i=2}^{\tau} p(x_i|x_{i-1}, \theta). \tag{3}$$

We consider three approaches for analysing these models. The first considers inference based on a single observation, and uses importance sampling (IS) to approximate (1) and get a weighted sample from (2). The key to efficiently implementing IS is the choice of proposal density, and for many models we can write the optimal proposal density in terms of transitions of a Markov process (see e.g. Stephens and Donnelly, 2000). The key idea is that it is much easier to construct approximations for these transitions than for the joint proposal density of $\mathbf{x}$. We demonstrate the application of these ideas for diffusion models, continuous-time Markov processes, and for inference in population genetics.

The latter two approaches consider inference based on a set of observations $\mathbf{y} = (y_1, \ldots, y_n)$ which can be thought of as being made over time. The first of these approaches is the Forward-Backward algorithm (Scott, 2002), which enables exact calculation of (1) and simulation from (2) when it can be applied. We show some of the range of problems for which the Forward-Background algorithm can be applied, and give an example of its use for analysing a multiple changepoint model of recombination in *Salmonella*. The last approach is that of Sequential

Monte Carlo (SMC), also know of particle filters. There is an extensive literature on SMC starting with the papers of Gordon et al. (1993) and Kong et al. (1994), particularly for online applications (see Doucet et al., 2001, for applications). We give a brief overview of some of this work, and show how ideas from SMC can be applied to both the Forward-Backward and IS algorithms considered previously.

The examples we consider are a combination of continuous and discrete time models. We have used subscript i in the discrete time case; and subscript t in the continuous time case. The examples are included to give insight into how these methods can be applied in practice, and it is possible to read the paper while omitting one or more of the examples. However some of the examples introduce novelty in terms of the specific area of application. In particular Example 1 proposes a new form of proposal distribution for the evalutation of likelihood for diffusion models; while Example 2 gives the first application of IS methods to conditioned simulation from a discrete-state, continuous-time Markov process.

2 Importance Sampling

Importance sampling (IS) is a standard Monte Carlo technique for estimating integrals (see e.g. Ripley, 1987). For example, consider esimating $I = \int g(\mathbf{x})\mathrm{d}\mathbf{x}$. If we have a density $q(\mathbf{x})$ such that $q(\mathbf{x}) > 0$ whenever $g(\mathbf{x}) > 0$ then

$$I = \int \frac{g(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x})\mathrm{d}\mathbf{x} = \mathrm{E}_q(w(\mathbf{X})),$$

where $w(\mathbf{X}) = g(\mathbf{X})/q(\mathbf{X})$ and the final expectation is with respect to $q(\mathbf{x})$. Thus a natural estimator of I is obtained by (i) sampling M iid draws from $q(\mathbf{x})$, which we denote $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}$; and (ii) estimating I by

$$\hat{I} = \frac{1}{M} \sum_{j=1}^{M} w(\mathbf{x}^{(j)}).$$

This estimator is consistent as $M \to \infty$; and, assuming $\mathrm{E}_q(w(\mathbf{X})^2) < \infty$, its variance is $\mathrm{Var}_q(w(\mathbf{X}))/M$.

The efficiency of the method crucially depends on the choice of $q(\mathbf{X})$. For our application it is appropriate to assume that $g(\mathbf{X}) \geq 0$ for all $\mathbf{X}$, in which case the optimal proposal distribution is $q_{\mathrm{opt}}(\mathbf{X}) = g(\mathbf{X})/I$ and produces an estimator with zero variance. It is not normally possible to simulate from $q_{\mathrm{opt}}(\mathbf{X})$, but this result can guide the choice of proposal. Note that, particularly in high dimensions, it is easy to choose proposal distributions for which $\hat{I}$ has infinite variance. This can occur if the proposal distribution has lighter tails than the optimal proposal; and for this reason it is common to choose a heavy-tailed proposal distribution.

For our application $g(\mathbf{x}) = p(\mathbf{x}|\theta)p(\mathbf{y}|\mathbf{x}, \theta)$, and $I = p(\theta|\mathbf{y})$, the likelihood. Before looking at some examples, and how to choose a suitable proposal distribution, we make three important comments. Firstly, IS does not only produce an estimate of the likelihood, it produces a weighted sample, $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)})$ with weights $(w(\mathbf{x}^{(1)}), \ldots, w(\mathbf{x}^{(M)}))$, that approximates the conditional distribution $p(\mathbf{x}|\mathbf{y}, \theta)$. Thus any expectation of the form $\int h(\mathbf{x})p(\mathbf{x}|\mathbf{y}, \theta)\mathrm{d}\mathbf{x}$ can be approximated by

$$\frac{\sum_{j=1}^{M} h(\mathbf{x}^{j})w(\mathbf{x}^{(j)})}{\sum_{j=1}^{M} w(\mathbf{x}^{(j)})}.$$

Secondly, it is possible to use IS to produce smooth estimates of the likelihood curve as a function of θ, rather than just estimate the likelihood independently at a fixed value of θ (or independently at a set of θ values). This can be seen by viewing the Monte Carlo estimator

$$\hat{I}(\theta) = \frac{1}{M}\sum_{j=1}^{M}\frac{p(\mathbf{x}^{(j)}|\theta)p(\mathbf{y}|\mathbf{x}^{(j)},\theta)}{q(\mathbf{x}^{(j)})}$$

as a function of θ. Practically this involves using the same proposal distribution and sample of $\mathbf{x}$ values to calculate the estimate of the likelihood for all θ values. There are considerable theoretical and practical advantages of using such smooth estimates when estimating θ via maximum likelihood (see Beskos et al., 2007; Pitt, 2007). Designing a good proposal distribution to estimate the likelihood at a range of θ values can be challenging (Stephens, 1999); though ideas from bridge sampling (Meng and Wong, 1996) can be used (Fearnhead and Donnelly, 2001).

Finally, the accuracy of the IS estimator can itself be estimated by looking at the variability of the weights. A common way of doing this is to use the Effective Sample Size of Liu (1996). This is defined as

$$\mathrm{ESS} = \frac{\left(\sum_{j=1}^{M} w(\mathbf{x}^{(j)})\right)^2}{\sum_{j=1}^{M} w(\mathbf{x}^{(j)})^2},$$

and its interpretation is that inference based on the weighted sample of size M, will be approximately as accurate as one based on a independent sample of size ESS. In some situations the estimated ESS value can be misleading: see the comments in Stephens and Donnelly (2000) for further discussion of this.

We now consider how to choose, or design, an efficient proposal density. We consider two cases, both where we have observations of a stochastic process at some time-point. In the first case, we also know the initial state of the process, and our proposal is based on forward simulation of the stochastic process from this initial state to the observed state at a later time. For the second case we have no initial state for the process, instead the assumption is that the stochastic process is at stationarity. In this case our approach is to simulate the stochastic process backwards in time from the observed state. More specific details and examples are considered below.

2.1 Choosing the Proposal Density: Forward Simulation

Assume that our model for the missing data satisfies (3), and that $\mathbf{Y}$ depends on $\mathbf{x}$ just through the final value x_τ. We have that the optimal proposal density is

$$q_{\text{opt}}(\mathbf{x}) = p(x_1|\theta,\mathbf{y})\prod_{i=1}^{\tau-1} p(x_{i+1}|x_i,\theta,\mathbf{y}), \tag{4}$$

where in this case $p(x_1|\theta,\mathbf{y}) \propto p(x_1|\theta)p(\mathbf{y}|x_1,\theta)$ and $p(x_{i+1}|x_i,\theta,\mathbf{y}) \propto p(x_{i+1}|x_i,\theta)p(\mathbf{y}|x_{i+1},\theta)$. (Note that in general the optimal proposal distribution will depend on θ.) Thus to design an efficient proposal distribution we only need to construct good approximations to the probabilities $p(\mathbf{y}|x_i,\theta)$.

Example 1: Discretely observed diffusion

Consider the problem of inference for a d-dimensional diffusion which is observed at discrete time points. The dynamics of the diffusion is specified by the stochastic differential equation

$$\mathrm{d}\mathbf{X}_t = \mu_\theta(\mathbf{X}_t)\mathrm{d}t + \sigma_\theta(\mathbf{X}_t)\mathrm{d}B_t,$$

where B_t is m-dimensional Brownian motion, the drift $\mu_\theta(\mathbf{X}_t)$ is a d-dimensional vector and the volatility $\sigma_\theta(\mathbf{X}_t)$ is a $d \times m$ dimensional matrix. In the following we consider inference for the likelihood for a single value of θ and drop the dependence on θ in our notation. (Note that producing smooth estimates of the likelihood curve can be non-trivial if the volatility depends on θ; we do not pursue this, but see for example Roberts and Stramer, 2001; Beskos et al., 2006, .)

We condition on $\mathbf{x}_0$ and consider a single observation $\mathbf{y} = \mathbf{x}_T$. The likelihood is just the transition density $p(\mathbf{x}_T|\mathbf{x}_0)$. Note that extension to multiple observations is straightforward, as the likelihood is just a product of transition densities. In general the transition density of a diffusion is intractable, and a common approach to analyse diffusions is by approximating the diffusion by a discrete-time Markov process (though see Beskos et al., 2006; Fearnhead et al., 2007, for methods where such a time-discretisation is not necessary). Thus we choose an integer n, let $h = T/n$, and define a process $\mathbf{X}_0, \mathbf{X}_h, \mathbf{X}_{2h}, \dots$ where

$$\mathbf{X}_{(i+1)h}|\mathbf{x}_{ih} = \mathbf{x}_{ih} + \mu(\mathbf{x}_{ih})h + h^{1/2}\sigma(\mathbf{x}_{ih})\mathbf{Z}_i,$$

where $\mathbf{Z}_i$ is a vector of m independent standard normal random variables. These dynamics are obtained from an Euler approximation to the SDE for $\mathbf{X}_t$ (Kloeden and Platen, 1992). This is a time-homogeneous Markov process, and we denote its transition density by $p_h(\cdot|\cdot)$.

Under this approximation, we have that $\mathbf{x} = (\mathbf{x}_h, \dots, \mathbf{x}_{(n-1)h})$ and $\mathbf{y} = \mathbf{x}_T$. Furthermore $p(\mathbf{y}|\mathbf{x}) = p_h(\mathbf{y}|\mathbf{x}_{(n-1)h})$ and $p(\mathbf{x}) = \prod_{i=1}^{(n-1)} p_h(\mathbf{x}_{ih}|\mathbf{x}_{(i-1)h})$. We will use IS to estimate the likelihood $p(\mathbf{y}|\mathbf{x}_0)$, and look at the effect of different proposal densities. For a proposal density $q(\mathbf{x})$, the importance sampling weight will be $p(\mathbf{x})p(\mathbf{y}|\mathbf{x})/q(\mathbf{x})$. In discussing and designing proposal densities we will work with the discrete-time Euler approximation above; however a more rigorous approach is to design proposal laws for the continuous time-process, before discretising time to implement the method. Such an approach is possible by working with the conditioned SDE, which can be obtained using the theory of Doob's h-transforms (Rogers and Williams, 2000).

The first attempt to use IS to estimate the likelihood for discretely observed diffusions was by Pedersen (1995), who used the proposal $q(\mathbf{x}) = p(\mathbf{x})$. More recently, Durham and Gallant (2002) proposed using a proposal distribution motivated by (4). The idea is to approximate $p(\mathbf{y}|\mathbf{x}_{ih})$ using an Euler approximation to the SDE. If we let $\Delta = (T - (i+1)h)$ denote the further time to T then we use

$$\mathbf{X}_T|\mathbf{x}_{(i+1)h}, \mathbf{x}_{ih} = \mathbf{x}_{(i+1)h} + \mu(\mathbf{x}_{(i+1)h})\Delta + \Delta^{1/2}\sigma(\mathbf{x}_{ih})\mathbf{Z}_i.$$

Note that this is slightly different from the Euler approximation as we have used $\sigma(\mathbf{x}_{ih})$ rather than $\sigma(\mathbf{x}_{(i+1)h})$. This is purely to simplify the resulting proposal distribution, and makes negligible difference for sufficiently small h. Thus the approximation to $p(\mathbf{y}|\mathbf{x}_{(i+1)h})$ is Gaussian, and it is straight forward to combine this approximation to the likelihood with the Gaussian transition density $p_h(x_{(i+1)h}|x_{ih})$. The resulting proposal is of the form $\prod_{i=0}^{n} \tilde{q}(\mathbf{x}_{(i+1)h}|\mathbf{x}_{ih})$ where $\tilde{q}(\mathbf{x}_{(i+1)h}|\mathbf{x}_{ih})$ is the pdf of a Gaussian distribution with mean $\mathbf{x}_{ih} + h(\mathbf{x}_T - \mathbf{x}_{ih})/(T - ih)$ and variance $h\Delta\sigma(\mathbf{x}_{ih})\sigma(\mathbf{x}_{ih})^T/(\Delta + h)$.

To demonstrate the advantage of designing a suitable proposal distribution for this problem, consider the following simple 1-dimensional diffusion:

$$\mathrm{d}X_t = (5 - X_t)\mathrm{d}t + X_t^{1/2}\mathrm{d}B_t. \tag{5}$$

This is a specific example of the CIR diffusion (Cox et al., 1985). Its transition density is known, but we do not use this fact in the following. We will consider estimating the transition density with $X_0 = 4.9$ and $X_T = 5.0$, for a range of T and h values.

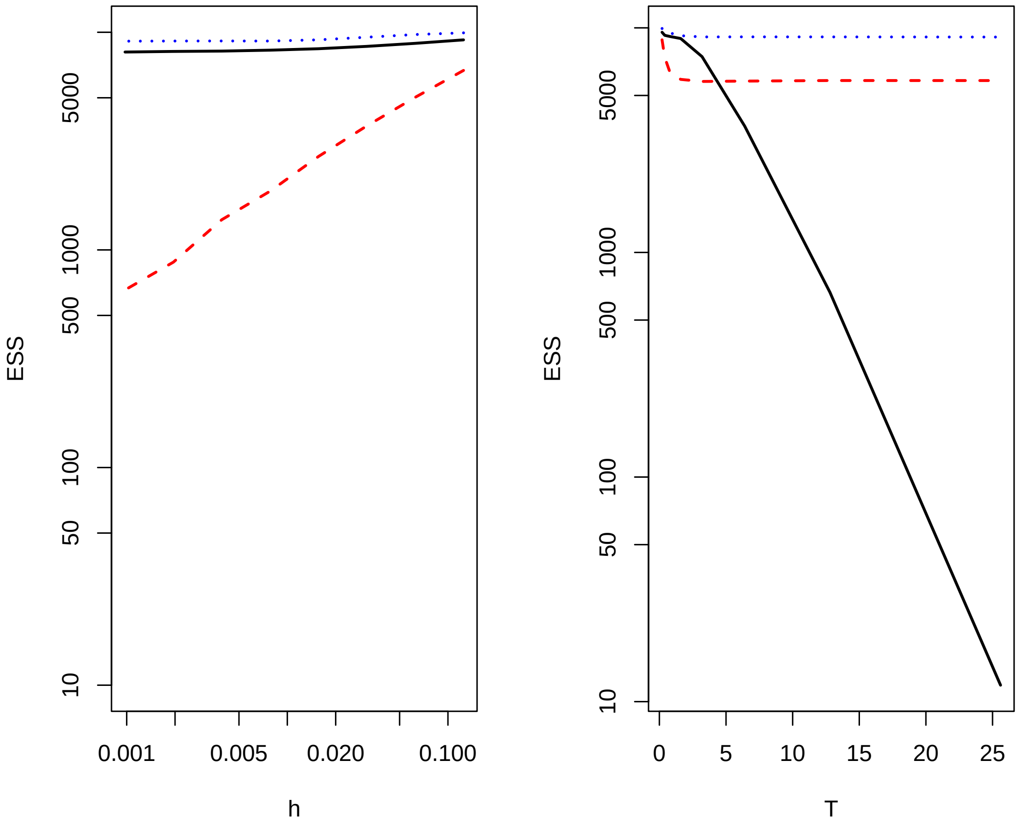

Figure 1: The efficiency of three IS proposals for estimating the transition density of a diffusion: (black full line) the method of Durham and Gallant;(red dashed line) the method of Pedersen; and (blue dotted line) the new proposal. Both figures are for $X_T = 5.0$ and $X_0 = 4.9$. The left-hand plot shows the ESS of the proposals for $T = 1$ and different values of h; the right-hand plot shows the ESS of the proposals for $h = 0.1$ and different values of T.

Firstly we fixed $T = 1$ and varied h in factors of 2 between $1/1024$ and $1/8$ (see left-hand plot in Figure 1). Here we notice that the Durham and Gallant proposal is robust to varying h, whereas the method of Pedersen performs poorly when h is small. This contrasting behaviour of the Pedersen and Durham and Gallant method can be shown to hold generally (see Stramer and Yan, 2007; Stramer, 2007; Delyon and Hu, 2006).

Secondly we fixed $h = 0.1$ and varied T in factors of 2 between 0.1 and 25.6 (see right-hand plot in Figure 1). Here we notice that the method of Pedersen is robust to increases in T, whereas the ESS of the Durham and Gallant proposal decreases at an exponential rate as T increases. The reason for this is that for a stationary diffusion like the CIR model, if $T - t$ is large then the dynamics of X_t conditional on X_T will be close to the dynamics of the unconditioned process. The Pedersen proposal is based on simulating X_t from this unconditioned process, and therefore does not deteriorate as T increases. By comparison the Durham and Gallant proposal takes account of the X_T value through an Euler approximation of $p(X_T|X_t)$, and this Euler approximation is poor for large $T-t$, and thus it gives a poor proposal for the dynamics of X_t when $T - t$ is large.

We can improve on the Durham and Gallant proposal by obtaining a better approximation to $p(x_T|x_t)$. For a geometrically ergodic diffusion, with stationary distribution $\pi(x)$, we have that

$||p(x_T|x_0) - \pi(x_T)|| < A\exp\{-\rho T\}$ for some constants A and ρ. Thus we can approximate

$$\hat{p}(x_T|x_t) = \exp\{-\rho(T-t)\}p_\Delta(x_T|x_t) + (1 - \exp\{-\rho(T-t)\})\pi(x_T),$$

where $p_\Delta(x_T|x_t)$ is the transition density of the Euler approximation over a time-interval $\Delta = T-t$. Now whilst ρ and $\pi(X_T)$ are unknown, this approximation suggests a proposal distribution which is a mixture

$$q(x_{(i+1)h}|x_{ih}) \propto \exp\{-\rho\Delta\}\tilde{q}(x_{(i+1)h}|x_{ih}) + B(1 - \exp\{-\rho\Delta\})p(x_{(i+1)h}|x_{ih})$$

where B and ρ are constants, $\tilde{q}(x_{(i+1)h}|x_{ih})$ is the Durham and Gallant proposal and $p(x_{(i+1)h}|x_{ih})$ is the transition of the unconditioned diffusion.

We tested this on the CIR diffusion above. We chose $B = 1$ and $\rho = 1$. The results show that the resulting proposal performs well for both small h and large T, and is uniformly better than both alternative proposals.

One general lesson from this example is that while using the optimal proposal is generally intractable, it is possible to use knowledge about the underlying process to learn about features of the optimal proposal, and these can then be used to design efficient proposal densities. We believe that proposals based on the argument above will have good performance for a range of geometrically ergodic diffusions.

Example 2: Discretely Observed Lokta-Volterra Process

We now consider a related example, namely that of a continuous-time discrete-valued Markov process. Here we will focus on a specific example, but the ideas can be generalised. The example we consider is the Lokta-Volterra process, where $\mathbf{X}_t = (X_t^{(1)}, X_t^{(2)})$ both $X_t^{(1)}$ and $X_t^{(2)}$ can take values in the non-negative integers, and they represent the number of prey and predator respectively.

The model we use comes from Boys et al. (2007). There are three possible transitions for the process. We denote $\lambda(\mathbf{x}, \mathbf{x}')$ to be the rate of transition from $\mathbf{x}$ to $\mathbf{x}'$, with

$$\lambda(\mathbf{x}, \mathbf{x}') = \left\{ \begin{array}{cl} \alpha x^{(1)} & \text{if } x'^{(1)} = x^{(1)} + 1 \text{ and } x'^{(2)} = x^{(2)}, \\ \beta x^{(1)} x^{(2)} & \text{if } x'^{(1)} = x^{(1)} - 1 \text{ and } x'^{(2)} = x^{(2)} + 1, \\ \gamma x^{(2)} & \text{if } x'^{(1)} = x^{(1)} \text{ and } x'^{(2)} = x^{(2)} - 1, \end{array} \right.$$

where α, β, and γ are positive constants. (All other transitions have rate 0.) This type of model is used for gene regulatory networks (Boys et al., 2007; Golightly and Wilkinson, 2005). The difficulty with analysing this model directly has led to the use of its diffusion approximation instead (Golightly and Wilkinson, 2005, 2006). The diffusion approximation is a two dimensional diffusion process, $\tilde{\mathbf{X}}$, with drift and instantaneous variance given by

$$\left(\begin{array}{c} \alpha\tilde{X}^{(1)} - \beta\tilde{X}^{(1)}\tilde{X}^{(2)} \\ \beta\tilde{X}^{(1)}\tilde{X}^{(2)} - \gamma\tilde{X}^{(2)} \end{array} \right), \text{ and } \left(\begin{array}{cc} \alpha\tilde{X}^{(1)} + \beta\tilde{X}^{(1)}\tilde{X}^{(2)} & -\beta\tilde{X}^{(1)}\tilde{X}^{(2)} \\ -\beta\tilde{X}^{(1)}\tilde{X}^{(2)} & \beta\tilde{X}^{(1)}\tilde{X}^{(2)} + \gamma\tilde{X}^{(2)} \end{array} \right) \text{ respectively.} \quad (6)$$

See Wilkinson (2006) for more details. The advantage of working with this approximation is that the ideas discussed in Example 1 can be used. Here we see if we can perform inference without resorting to this approximation.

We denote the transition density of the Lokta-Volterra process over time t by $p_t(\cdot|\cdot)$. We assume we know the state of the process at time 0, $\mathbf{x}_0$, and that we observe the process without error at time T. Thus $\mathbf{y} = \mathbf{x}_T$, and the likelihood is just the transition density $p_T(\mathbf{x}_T|\mathbf{x}_0)$. We estimate this using importance sampling. Our proposal process will be a continuous time Markov process. Let $\Delta = T - t$ be the further time from t until the observation. We can generalise (4), which gives that the optimal proposal has transition rates, which we denote $\lambda_t(\mathbf{x}, \mathbf{x}')$, where

$$\lambda_t(\mathbf{x}, \mathbf{x}') = \lim_{\delta t \to 0} \frac{\Pr(\mathbf{X}_{t+\delta t} = \mathbf{x}' | \mathbf{X}_t = \mathbf{x}, \mathbf{x}_T)}{\delta t} = \lim_{\delta t \to 0} \frac{p_{\delta t}(\mathbf{x}'|\mathbf{x})}{\delta t} \frac{p_{\Delta - \delta t}(\mathbf{x}_T|\mathbf{x}')}{p_\Delta(\mathbf{x}_T|\mathbf{x})}.$$

Thus we get $\lambda_t(\mathbf{X}, \mathbf{X}') = \lambda(\mathbf{X}, \mathbf{X}')p_\Delta(\mathbf{x}_T|\mathbf{x}')/p_\Delta(\mathbf{x}_T|\mathbf{x})$. Thus to get a good approximation we need only get an approximation to the transition density of the process.

One simple idea we have tried is to approximate the transition density using the Euler approximation to the diffusion approximation (6), which is the idea used in the Durham and Gallant proposal in Example 1. There are two difficulties with this approach which need to be overcome.

The first is simulating from the resulting time-inhomogeneous process. A simple way around this is to choose a time-homogeneous proposal which approximates the above. Thus at an event, say at time t, (i) we calculate our approximation to $p_\Delta(\mathbf{x}_T|\mathbf{x})$ for the four relevant values of $\mathbf{x}$ (namely $\mathbf{x}_t$, and the values of $\mathbf{x}$ obtained after the three possible transitions of the process); (ii) calculate the resulting approximations to $\lambda_t(\mathbf{x}, \mathbf{x}')$ that we get for the three possible transitions; and (iii) simulate the time and type of the next event assuming a continuous-time process with these rates fixed.

The second is that the diffusion approximation breaks down for very small time intervals: so we need a separate approximation for when $\Delta (= T - t)$ is sufficiently small. However there is a simple approximation based upon (i) fixing the rates of the three events (to $r_1 := \alpha x_t^{(1)}$, $r_2 := \beta x_t^{(1)} x_t^{(2)}$ and $r_3 := \gamma x_t^{(2)}$ respectively); and (ii) summing up only the probabilities of the paths from $\mathbf{x}_t$ to $\mathbf{x}_T$ that contain the fewest number of events. Thus if (n_1, n_2, n_3) denotes the number of events of each type in such a path (which is well-defined), then we approximate $p_\Delta(\mathbf{x}_T|\mathbf{x}_t)$ by

$$\left(\frac{n!}{n_1!n_2!n_3!}\right) \left[\prod_{i=1}^{3} \left(\frac{r_i}{r_1 + r_2 + r_3}\right)^{n_i}\right] \left(\frac{[(r_1 + r_2 + r_3)\Delta]^n}{n!} \exp\{-(r_1 + r_2 + r_3)\Delta\}\right),$$

where $n = n_1 + n_2 + n_3$. This equation simplifies, but here the last term is the probability of n events in time Δ, the second is the conditional probability that a specific ordered set of n events will be of the correct type, and the first is the number of arrangements of the n events. This gives a time-inhomegeous proposal process. To make this explicit we substitute $\Delta = T - t$. Under the proposal process we have that if $n_j > 0$ then the rate of an event of type j is $n_j/(T-t)$; whereas if $n_j = 0$ then the rate of an event of type j is $r_1 r_2 r_3 (T-t)^2 / \prod_{i=1}^{3}(n_i + 1)$. Direct simulation from this time-inhomogeneous process is straightforward. Our approach is to simulate from this process when $\Delta < \epsilon$ for a suitably chosen ϵ. (A simple choice of ϵ is the inverse of the total rate of events when the state is $\mathbf{x}_T$.)

We omit full details of the IS algorithm, though code in R implementing this procedure is available from `www.maths.lancs.ac.uk/∼fearnhea`. Our aim has been to try and show how a suitable proposal process can be obtained. Instead we just give numerical results for one example. We fixed $\alpha = 2$, $\beta = 1/20$ and $\gamma = 1.5$. An example simulated path is shown in Figure 2. We fixed $\mathbf{x}_0 = (68, 7)$ and considered estimating $p_t(\mathbf{x}_t|\mathbf{x}_0)$, for different values of t between 0.02 and 0.64, and the values of $\mathbf{x}_t$ used were taken from the simulated path shown in Figure 2.

We implemented our procedure using $M = 1000$. The ESS values of the estimates are shown in Table 1. We also show the gain in efficiency over a naive Monte Carlo estimate, which proposes from the prior and gives paths a weight of 1 if the value of $\mathbf{x}_t$ corresponds to the observed

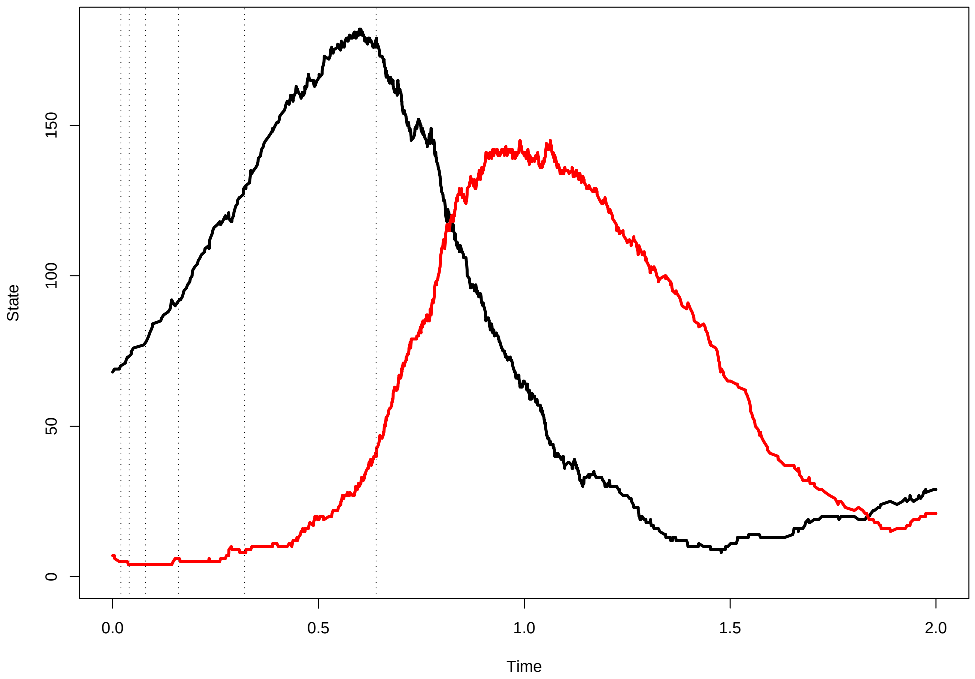

Figure 2: Simulated path for the Lokta-Volterra model. Prey ($x_t^{(1)}$) and predator ($x_t^{(2)}$) paths are shown in black and red respectively. The time-points used in the simulation study of the IS estimator are shown by vertical dashed lines.

t	0.02	0.04	0.08	0.16	0.32	0.64
ESS	340	398	380	205	9	5
Rel. Eff.	41	140	86	20	8	35

Table 1: ESS and Relative Efficiency of the IS procedure for estimating $p_t(\mathbf{x}_t|\mathbf{x}_0)$ for different t for the Lokta-Volterra example of Figure 2. Results based on 1,000 samples.

value, and a weight of 0 otherwise. This gain in efficiency is ESS$/(Mp_t(\mathbf{x}_t, \mathbf{x}_0))$, where ESS is the ESS of the IS procedure, and $Mp_t(\mathbf{x}_t, \mathbf{x}_0)$ is the "effective sample size" of the naive Monte Carlo estimator (i.e. the expected number of paths which coincide with the observed value).

The method has a high ESS for values of t up to $t = 0.16$; but the IS method's performance deteriorates for larger t. This is similar to the qualitative results for the Durham and Gallant IS method in Example 1, and the reason will again be that the approximations used when designing the proposal density deteriorate for large t. As in Example 1, it may be possible to improve the approximations so that the method can estimate the transition density for large t.

Our new IS procedure has larger CPU cost, and for our simulations takes about 50 times longer to propose a single path than the naive Monte Carlo method (though a more efficient implementation of the method may reduce this). After taking this into account we see that the IS procedure is still more efficient for $t = 0.04$ and $t = 0.08$. The naive Monte Carlo estimator will become inefficient at an exponential rate as the dimension of the state-space increases, whereas the performance of the IS method should be more robust to such an increase. Thus we would expect the IS procedure to become relatively more efficient for higher dimensional problems.

2.2 Choosing the Proposal Density: Backward Simulation

We now consider inference for stochastic processes given a single observation $\mathbf{x}_0$. We assume the underlying process is at stationarity, with stationary distribution $\pi(\cdot)$, and our interest is to estimate $\pi(\mathbf{x}_0)$. This is a common problem, and the methods we describe below can be applied to a range of problems in population genetics (e.g. Griffiths and Tavaré, 1994; Stephens and Donnelly, 2000; Bahlo and Griffiths, 1998; Fearnhead and Donnelly, 2001, 2002) and elsewhere (see Chen et al., 2005, for some examples). (See Bhattacharya et al., 2007, for an alternative approach to this problem.)

The IS method we consider assumes that there is a set of values of the state χ for which $\mathbf{x} \in \chi$ means that $\pi(\mathbf{x})$ can be evaluated analytically. We will focus on discrete-time Markov processes (though the idea can be extended to continuous-time processes). The basic idea is to simulate the path of the process back in time from $\mathbf{x}_0$ until the time τ when $\mathbf{x}_i$ is first in χ. (For this method to be practicable, we are implicitly assuming that the expectation of the stopping time τ is finite.) For notational convenience in what follows we will reverse time, so that a time $i > 0$ will relate to a time i discrete time-steps in the past.

Now if $p(\mathbf{x}'|\mathbf{x})$ denotes the transition density of the underlying stochastic process, then we have that

$$\pi(\mathbf{x}_0) = \sum_{\tau} \int \pi(\mathbf{x}_\tau) \prod_{i=0}^{\tau-1} p(\mathbf{x}_i|\mathbf{x}_{i+1}) \mathrm{d}\mathbf{x}_{1:\tau},$$

where $\mathbf{x}_{1:\tau} = (\mathbf{x}_1, \dots, \mathbf{x}_\tau)$. The sum is over all possible values of the stopping time; and the integral is over all paths that produce such a stopping time. Thus we can simulate paths $\mathbf{x}^{(j)}_{1:\tau_j}$, for $j = 1, \dots, M$, from a Markov proposal distribution, with transiton density $q(\mathbf{x}_{t+1}|\mathbf{x}_t)$; and produce an IS estimate of $\pi(\mathbf{x}_0)$:

$$\frac{1}{M} \sum_{j=1}^{M} \pi(\mathbf{x}_{\tau_j}) \left(\prod_{i=0}^{\tau_j - 1} \frac{p(\mathbf{x}_i|\mathbf{x}_{i+1})}{q(\mathbf{x}_{i+1}|\mathbf{x}_i)} \right). \tag{7}$$

Stephens and Donnelly (2000) showed that the optimal proposal distribution for this class of problem is

$$q_{\text{opt}}(\mathbf{x}_{i+1}|\mathbf{x}_i) = p(\mathbf{x}_i|\mathbf{x}_{i+1})\pi(\mathbf{x}_{i+1})/\pi(\mathbf{x}_i). \tag{8}$$

Substitution of this optimal proposal into (7) shows directly that this produces an IS estimate with zero variance. Note that this optimal proposal can be interpreted as the transition for the time-reversed process. The optimal proposal cannot be used directly as it requires knowledge of the stationary distribution, but the advantage of this formulation is that designing a good proposal distribution will only require approximating ratios of the form $\pi(\mathbf{x}_{i+1})/\pi(\mathbf{x}_i)$. For specific examples of how to approximate this ratio in different applications see Stephens and Donnelly (2000); Fearnhead and Donnelly (2001); De Iorio and Griffiths (2004a) and De Iorio and Griffiths (2004b).

Example 3: Temporally-sampled data in population genetics

The above IS approach has proven popular within population genetics, which is from where we take this example. We consider inference for data collected at a single genetic locus taken from randomly sampled chromosomes. We assume a constant population size, with random-mating, and model the data via the coalescent (Kingman, 1982).

The coalescent is a stochastic process that describes the ancestry of a sample. As the process goes back in time, the number of distinct ancestors of the sample decreases, until there is a single common ancestor. (For fuller details and background see Donnelly and Tavaré (1995), Wakeley (2007) or the Introduction of Stephens and Donnelly (2000).) For most mutation models for the genetic locus, it is possible to calculate the stationary distribution of the type of the common ancestor of the sample, but not the distribution of a sample of size greater than one. Thus the above IS method can be used to approximate the probability of a given sample, with the underlying process being the coalescent, and the stopping time being the first time there is a single ancestor for the sample.

Here we consider the infinite alleles mutation model of Kimura and Crow (1964). This model assumes that each mutation produces a distinct genetic type. The data records which chromosomes are identical to each other at the locus of interest - but contains no information about the degree of difference of genetically distinct chromosomes. This is an appropriate model for example for MLST data for bacteria (Maiden et al., 1998). This is an example of a model for which the stationary distribution of any sample is known – but we will extend it to a more complicated situation below. The model is parameterised by a mutation rate θ and the product of the effective population size and generation time $N_e g$; the latter governs the time-scale on which the population evolves.

Assume data collected at a single time point, $\mathbf{y}_0$. We assume that there are K_0 genetic types in our sample, labelled $1, \dots, K_0$. Our data will record how many chromosomes in the sample carry each type. If we summarise our data $\mathbf{y}_0$ by the number of distinct genetic types K_0, the total sample size n_0, and the number of individuals of each type, n_0^k for $k = 1, \dots, K_0$, then the stationary distribution is given by

$$\pi(\mathbf{y}_0) = \theta^{K_0} \frac{\Gamma(\theta)}{\Gamma(n_0 + \theta)} \frac{n_0!}{\prod_{k=1}^{K} n_0^{(k)}}, \tag{9}$$

Note that this does not depend on $N_e g$.

We define the coalescent process in continuous time, with $\mathbf{x}_0 = \mathbf{y}_0$. The state of the coalescent at time t, $\mathbf{x}_t$ will consist of the types of the ancestors of the sample at time t in the past. The dynamics of the coalescent process is given in terms of event times, and transitions at these event time. We will let t_i denote the time of the ith event back in time (with $t_0 = 0$). The distribution of the inter-event time $T_{i+1} - T_i$ are exponentially distributed with rate $n_i(n_i + \theta - 1)/(2N_e g)$, where n_i are the number of ancestors of the sample at time t_i (for our application this will be $n_i = n_0 - i$).

At an event time, there are two types of events, called coalescences and mutations. The former refers to a pair of the ancestors at time t themselves sharing a common ancestor, and results in a transition to $\mathbf{x}_t$ which removes one copy of the genetic type. The latter refer to points when an ancestor underwent a mutation. For our mutation model, we do not need to continue to trace the ancestry at these mutation events (see Fearnhead, 2002a), so these events will also lead to the removal of one copy of genetic type from $\mathbf{x}_t$. As mutations are unique, mutations can only occur to types k for which there is just a single copy.

We will slightly abuse notation and define $\mathbf{x}_i$ to be $\mathbf{x}_{t_i}$ the state immediately after the ith event, and let n_i denote the number of ancestors in $\mathbf{x}_i$, and $n_i^{(k)}$ the number of these which are of type k. Then if $\mathbf{x}_i$ differs from $\mathbf{x}_{i+1}$ just by the additition of a *specific* extra ancestor of type l, we have that the forward transition probabilities are

$$p(\mathbf{x}_i|\mathbf{x}_{i+1}) = \begin{cases} \frac{(n_i^{(l)}-1)}{(n_i-1+\theta)} & \text{if } n_i^{(l)} > 1, \\ \frac{\theta}{(n_i-1+\theta)} & \text{otherwise.} \end{cases}.$$

All other transitions have probability 0.

Now if we define $\lambda_i = n_i(n_i + \theta - 1)/(2N_e g)$, then the probability of a given path back to a stopping time T, during which there were τ events, is

$$\pi(\mathbf{x}_T)\left(\prod_{i=0}^{\tau-1} p(\mathbf{x}_i|\mathbf{x}_{i+1})\lambda_i \exp\{-\lambda_i(t_{i+1}-t_i)\}\right)\exp\{-\lambda_\tau(T-t_\tau)\}. \tag{10}$$

For this model, we can calculate the optimal proposal for estimating $\pi(\mathbf{y}_0)$. Using the results from Stephens and Donnelly (2000) and Stephens (2000), we have that the optimal proposal has rates of transition from $\mathbf{x}_t$ to $\mathbf{x}'$ which are

$$\lambda(\mathbf{x}', \mathbf{x}_t) = n_t^{(l)}(n_t + \theta - 1)/(2N_e g), \tag{11}$$

for all transitions such that $\mathbf{x}'$ differs from $\mathbf{x}_t$ by the removal of an ancestor of type l. Note that these rate describe the conditional dynamics of the coalescent given $\mathbf{y}_0$ (Stephens and Donnelly, 2000).

We now consider the extension to analysing data sampled at multiple time-points. Such data is informative about $N_e g$ as well as θ, and the problem is motivated by the analysis of *Campylobacter jejuni* in Wilson and Fearnhead (2007). For work on the same problem, but assuming different types of genetic data see Drummond et al. (2002) and Drummond et al. (2005).

For simplicity we consider data collected at two time-points, the current time and a time T in the past. We denote the data at the two time-points by $\mathbf{y}_0$ and $\mathbf{y}_T$ respectively. We define $\pi_t(\mathbf{y}, \mathbf{z})$ to denote the probability under stationarity of observing two samples, of types $\mathbf{y}$ and $\mathbf{z}$, sampled a time interval t apart. We further denote $p_t(\mathbf{y}|\mathbf{z}) = \pi_t(\mathbf{y}, \mathbf{z})/\pi(\mathbf{z})$. Our interest lies in estimating $\pi_T(\mathbf{y}_0, \mathbf{y}_T)$. We will denote the size of the sample at time T by m, and the number of these which are of type k by $m^{(k)}$.

We will use the IS approach described above. The stopping time will be the fixed time T, as $\pi_0(\mathbf{x}_T, \mathbf{y}_T)$ is known: it is related tothe stationary distribution of a sample of type $(\mathbf{x}_T, \mathbf{y}_T)$:

$$\pi_0(\mathbf{x}_T, \mathbf{y}_T) = \pi(\mathbf{x}_T, \mathbf{y}_T)\frac{\prod_{k=1}^K n_t^{(k)}!m_t^{(k)}!}{\prod_{k=1}^K (n_t^{(k)} + m_t^{(k)})!}$$

where K is the total number of alleles in the combined sample $(\mathbf{y}_T, \mathbf{x}_t)$, and the final combinatorial factor accounts for the knowledge of how many of each allelic type are from each of the $\mathbf{y}_T$ and $\mathbf{x}_T$ samples. The probability of a path back to time t will now be of the form (10), but with $\pi(\mathbf{x}_T)$ replaced by $\pi_0(\mathbf{x}_T, \mathbf{y}_T)$. The optimal proposal distribution is again the conditioned coalescent process. An expression for the rates of this process are obtained by taking the rates conditioned just on $\mathbf{y}_0$, and then also conditioning on $\mathbf{y}_T$. By a similar argument to that of Example 2, at time t, for transitions which remove an ancestor of type l, the rate is

$$\tilde{\lambda}_\Delta(\mathbf{x}', \mathbf{x}_t) = \frac{n_t^{(l)}(n_t + \theta - 1)}{2N_e g}\frac{p_\Delta(\mathbf{y}_T|\mathbf{x}')}{p_\Delta(\mathbf{y}_T|\mathbf{x}_t)},$$

where $\Delta = T - t$. Note we have parameterised this rate function in terms of Δ the further time to time T.

The key to approximating this, is to approximate the final ratio. One approach is to consider the limiting behaviour of this. Firstly as $\Delta \to \infty$, the ratio will tend to 1 (as the data at time t and T will be independent). Secondly as $\Delta \to 0$ we get that $p_\Delta(\mathbf{y}_T|\mathbf{x}_t) \to \pi_0(\mathbf{y}_T, \mathbf{x}_t)/\pi(\mathbf{x}_t)$. Therefore

$$\lim_{\Delta\to 0+} \tilde{\lambda}_\Delta(\mathbf{x}', \mathbf{x}_t) = \begin{cases} \frac{n_t + m + \theta - 1}{2N_e g} & \text{if } n_t^{(l)} = 1 \text{ and } m^{(l)} = 0, \\ \frac{n_t^{(l)}(n_t + m + \theta - 1)(n_t^{(l)} - 1)}{2N_e g(n_t^{(l)} + m^{(l)} - 1)} & \text{if } n_t^{(l)} > 1, \end{cases}$$

for transitions which remove an ancestor of type l. All other transitions have rate 0; this includes the removal of a ancestor of type l when $n_t^{(l)} = 1$ and $m^{(l)} \geq 1$. In this last case, the transition would relate to a mutation event, yet as each mutation is unique this is not possible as there is at least one chromosome of type l in the sample at time T.

Thus this motivates defining a function $\gamma(\Delta)$ with $\gamma(0) = 1$ and $\gamma(\Delta) \to 0$ as $\Delta \to \infty$, and proposing from a proposal with rate

$$\hat{\lambda}_\Delta(\mathbf{x}', \mathbf{x}_t) = \begin{cases} \frac{n_t+\gamma(\Delta)m+\theta-1}{2N_e g} & \text{if } n_t^{(l)} = 1 \text{ and } m^{(l)} = 0, \\ \frac{n_t^{(l)}(n_t+\gamma(\Delta)m+\theta-1)(n_t^{(l)}-1)}{2N_e g(n_t^{(l)}+\gamma(\Delta)m^{(l)}-1)} & \text{if } n_t^{(l)} > 1. \end{cases}$$

Thus $\gamma(\Delta)$ governs the amount of dependence that the sample at time T has on the transitions of the proposal at time t. As the underlying population genetic model is geometrically ergodic (Donnelly and Kurtz, 1996), and the time-scale on which the population evolves is proportional to $N_e g$, a natural choice is $\gamma(\Delta) = \exp\{-c\Delta/(N_e g)\}$ for some c.

We evaluated the performance of this IS method on simulated data. We considered 4 scenarios, which varied the sample size at times 0 and T, n_0 and m respectively, and the mutation rate, θ. The scenarios are: (a) $n_0 = 100$, $m = 20$, $\theta = 3$; (b) $n_0 = 50$, $m = 20$, $\theta = 3$; (c) $n_0 = 100$, $m = 20$, $\theta = 1$; and (d) $n_0 = 100$, $m = 40$, $\theta = 3$. For each case we chose 10 values of T, varying from $0.025N_e g$ to $2.5N_e g$.

We considered three IS approaches; which correspond to different values of c in the $\gamma(u)$ function above. These were $c = 0$, the value used by Wilson and Fearnhead (2007); $c = \infty$, which corresponds to proposing from the distribution of the genealogy conditional just on $\mathbf{y}_0$, i.e. it ignores the information in $\mathbf{y}_T$; and $c = 4$, chosen after experimenting with a small number of values of c. Results are given in Figure 3, where we show the ESS for each IS method averaged across ten simulated data sets for each combination of scenario and T.

The performance of the choices $c = 0$ and $c = \infty$ are intuitive: choosing $c = \infty$ ignores the information in $\mathbf{y}_T$, and so performs well for larger T. By comparison $c = 0$ performs well for small T, but produces a poor IS proposal for large T as it places too much weight on $\mathbf{y}_T$. Our proposal density with $c = 4$ gives almost uniformly better performance than both these approaches, and is robust for both small and larger values of T.

3 The Forward-Backward Algorithm

We now change focus to situations where we observe data over time $\mathbf{y} = (y_1, \ldots, y_n)$. We will assume a state-space model, where we have an underlying state of interest which varies over time. We will let $\mathbf{x} = (x_1, \ldots, x_n)$ denote the values of the state at the time-points at which the observations are made. In the presentation we are assuming both scalar states and observations, but the ideas apply equally to vector valued states and/or observations. We assume that the state is a Markov model, whose dynamics are governed by a transition density $p(x_i|x_{i-1})$; and that conditional on x_i, the observation is independent of the state and observations at other times. The likelihood of a given observation is denoted by $p(y_i|x_i)$. Finally we will assume a known distribution for x_1, $p(x_1)$. In many applications, these densities will depend on unknown parameters, but we suppress this dependence here.

A key to analysing such a model is the calculation, or estimation, of the *filtering densities*, which are $p(x_i|\mathbf{y}_{1:i})$ for $i = 1, \ldots, n$, where $\mathbf{y}_{1:i} = (y_1, \ldots, y_i)$. These densities are the posterior distribution for x_i given the data to time i. A standard recursion relates the filtering densities

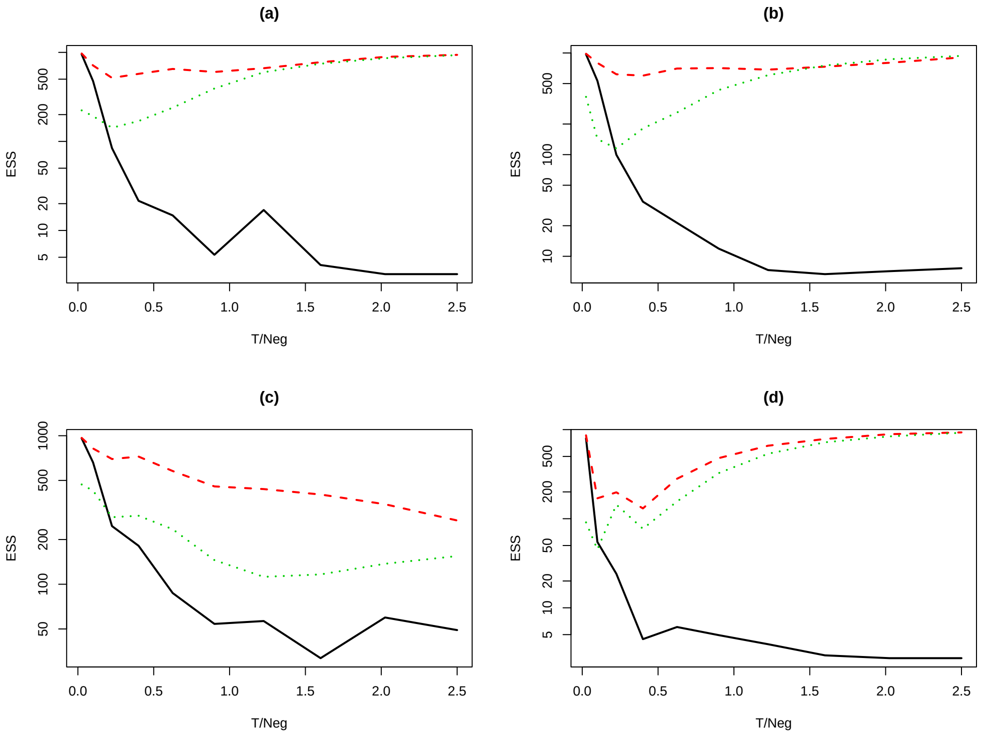

Figure 3: Average ESS values over ten simulated data sets for the three IS methods: $c = 0$ (black, full lines); $c = 4$ (red, dashed lines); and $c = \infty$ (green, dotted lines). Results are based on $M = 1,000$ in each case, and are shown for four scenarios (see text for details of these).

at successive time points:

$$p(x_i|\mathbf{y}_{1:i}) \propto p(y_i|x_i) \int p(x_i|x_{i-1})p(x_{i-1}|\mathbf{y}_{1:i-1})\mathrm{d}x_{i-1}, \tag{12}$$

for $i = 2, \dots, n$. The normalising constant of the right-hand side is just $p(y_i|\mathbf{y}_{1:i-1})$. We further have $p(x_1|y_1) \propto p(x_1)p(y_1|x_1)$, with the normalising constant being $p(y_1)$.

Being able to calculate the filtering densities, and the normalising constants of these recursions, is sufficient to (i) being able to calculate the likelihood as $p(\mathbf{y}) = p(y_1)\prod_{i=2}^{n} p(y_i|\mathbf{y}_{1:i-1})$; and (ii) being able to simulate from $p(\mathbf{x}|\mathbf{y})$, as

$$p(x_i|\mathbf{y}, \mathbf{x}_{i+1:n}) \propto p(x_i|\mathbf{y}_{1:i})p(x_{i+1}|x_i),$$

and thus we can simulate x_n from $p(x_n|\mathbf{y})$ and then recursively simulate x_i given $\mathbf{x}_{(i+1):n}$ for $i = n-1, \dots, 1$.

For most models, solving (12) analytically is not possible. Two important exceptions are for linear-Gaussian models when the solution is given be the Kalman Filter (Kalman and Bucy, 1961; West and Harrison, 1989); and for discrete states which can take only a finite number of values. This latter case is what we focus on here.

Assume that x_i takes values in $(1, \dots, K)$; then we can rewrite (12) as

$$p(x_i = j|\mathbf{y}_{1:i}) \propto p(y_i|x_i = j)\sum_{k=1}^{K} p(x_i = j|x_{i-1} = k)p(x_{i-1} = k|\mathbf{y}_{1:i-1}), \tag{13}$$

with

$$p(y_i|y_{1:i-1}) = \sum_{j=1}^{K} p(y_i|x_i = j)\sum_{k=1}^{K} p(x_i = j|x_{i-1} = k)p(x_{i-1} = k|\mathbf{y}_{1:i-1}),$$

and $p(x_i = j|\mathbf{y}, x_{i+1:n}) \propto p(x_i = j|\mathbf{y}_{1:i})p(x_{i+1}|x_i = j)$. All distributions are staightforward to simulate from, as they are on a finite space. These recursions are often known as the Forward-Backward algorithm, because they involve a forward pass through the data to calculate the filtering distributions and likelihood, followed by a backward simulation of the $\mathbf{x}$ from its conditional distribution.

The Forward-Backward algorithm has been used widely. It dates back to the work of Baum et al. (1970), where it is used within an EM algorithm. It has been rediscovered in various fields, and is known also by names such as the sum-product algorithm (Kschischang et al., 2001) and HMM algorithm; it is closely related to the Junction-Tree algorithm for graphical models (Cowell et al., 1999). It has been particularly important within speech recognition (Rabiner and Juang, 1986; Juang and Rabiner, 1991) and Bioinformatics (Durbin et al., 1998; Lander and Green, 1987; Felsenstein and Churchill, 1996). For an excellent review of the Forward-Backward algorithm within statistics, and in particular its use within MCMC methods, see Scott (2002).

In recent years there has been much research into extending the application of the Forward-Backward algorithm. One specific area is to continuous time processes (Fearnhead and Meligkotsidou, 2004), and in particular Markov modulated Poisson processes (Scott, 1999; Fearnhead and Sherlock, 2006). A second area is to that of inference for changepoint models. For the latter application, the link to the Forward-Backward algorithm is made apparent in Fearnhead (2006), though related approaches date to the work of Yao (1984) (see also Barry and Hartigan, 1992, 1993; Liu and Lawrence, 1999). Here we only focus on this latter application area, and base our presentation on the online algorithm of Fearnhead and Liu (2007).

The Forward-Backward algorithm can be applied to changepoint models which have a conditional independence property: given the position of a changepoint, the data before that changepoint is independent of the data after the changepoint. Such a model can be constructed in a hierachical manner as follows.

Firstly we model the changepoint positions via a Markov process. Here we focus on models where

$$\Pr(\text{next changepoint at } i|\text{changepoint at } j) = g(i-j).$$

Thus the probability mass function $g(\cdot)$ specifies the distribution of the length of regions between successive changepoints; we call these regions *segments* from now on.

Next we condition on m changepoints at times $\tau_1, \tau_2, \ldots, \tau_m$. We let $\tau_0 = 0$ and $\tau_{m+1} = n$, so our changepoints define $m+1$ segments, with segment k consisting of observations $\mathbf{y}_{\tau_k+1:\tau_{k+1}}$ for $k = 0, \ldots, m$. For a segment consisting of observations $\mathbf{y}_{j+1:i}$ we will have a set of unknown parameters, β say. We have a prior distribution, $\pi(\beta)$ for β, but assume that the parameters for this segment are independent of the parameters in other segments. Finally we define the marginal likelihood as

$$P(j,i) \quad = \quad \int p(\mathbf{y}_{j+1:i}|\beta)\pi(\beta)\mathrm{d}\beta, \tag{14}$$

and assume that these probabilities can be calculated for all $j < i$. This requires either conjugate priors for β, or the use of numerical integration.

Now we introduce the state at time i, x_i to be the time of the most recent changepoint prior to time i. We have that $x_i = i-1$ or $x_i = x_{i-1}$, corresponding to the presence or absence of a changepoint at time $i-1$. Furthermore $\mathbf{x}$ is a Markov process with

$$p(x_i = i-1|x_{i-1} = j) = \frac{g(i-1-j)}{\sum_{k=i-1}^{\infty} g(k-j)},$$

and $p(x_i = j|x_{i-1} = j) = 1 - p(x_i = i-1|x_{i-1} = j)$.

Using the conditional independence property, it can be shown that $p(y_i|x_i = j, \mathbf{y}_{1:i-1}) = P(j,i)/P(j,i-1)$, where $P(\cdot,\cdot)$ is the marginal likelihood defined above (14). Thus (13) becomes

$$p(x_i = j|\mathbf{y}_{1:i}) \propto \frac{P(j,i)}{P(j,i-1)} p(x_i = j|x_{i-1} = j) p(x_{i-1} = j|\mathbf{y}_{1:i-1})$$

for $j < i-1$, and

$$p(x_i = i-1|\mathbf{y}_{1:i}) \propto P(i-1,i) \sum_{j=0}^{i-2} p(x_i = i-1|x_{i-1} = j) p(x_{i-1} = j|\mathbf{y}_{1:i-1})$$

Thus, the filtering densities can be calculated recursively for this class of models, as in the Forward-Backward algorithm. The normalising constant of these equations is just $p(y_i|\mathbf{y}_{1:i-1})$. Furthermore, simulating from the joint distribution of changepoints is straightforward via backward simulation, with $p(x_i|\mathbf{y}, x_{i+1} = i) = p(x_i|\mathbf{y}_{1:i})$ and $x_i = j$ if $x_{i+1} = j$ for $j < i$.

The computational complexity of these recursions increases linearly with i, and thus there is a quadratic computational cost (and storage cost) for analysing a complete data set. The recursions hold conditional on knowing the hyperparameters of the distributions such as $\pi(\beta)$ and $g(\cdot)$; though Fearnhead (2006) show how the method can be efficiently implemented within an MCMC algorithm when hyperparameters are unknown; and Fearnhead and Vasileiou (2007) consider using an EM algorithm to estimate the hyperparameters. It is also straightforward to allow for model choice within segments (Fearnhead, 2005a) and also some types of dependence across segments (Fearnhead and Vasileiou, 2007).

Example 4: Analysis of Typhoid divergence

We demonstrate the method in an analysis of divergence data for *Samonella* Paratyphi A and *Salmonella* Typhi genomes. The data comes from Didelot et al. (2007), and consists of aligned sequences from the two genomes. In total there are 44 regions of aligned sequences, with a total length of 4.6Mb. For simplicity we give results below solely for analysis of the longest region, which is 606.2kb in length. We summarise the divergence of the two genomes in this region by (i) splitting the data into 6062 non-overlapping windows, each 100bp in length; and (ii) counting the number of nucleotide differences between the Paratyphi A and Typhi sequences in each window.

As in Didelot et al. (2007) we consider a changepoint model for the data. Such a model is appropriate as the divergence of the two sequences depends on how closely related the two strains of *Salmonella* are, and this will vary along the genome due to recombination (see Falush et al., 2006; Didelot et al., 2007, for more background). Didelot et al. (2007) performed inference using reversible jump MCMC; our aim is to show how simple it is to implement the Forward-Backward algorithm for this data.

We consider a more complex model than that of Didelot et al. (2007), through the use of an extra hierarchy. We allow for variability in divergence within segments by assuming a model where if $\mathbf{y}_{j+1:i}$ belong to a single segment then we have

$$\mathbf{y}_k \sim \text{Poisson}(\theta_k), \text{ where } \theta_k \sim \text{Gamma}(\alpha, \beta),$$

for $k = j+1, \ldots, i$, with the θ_ks being independent of each other. The inclusion of the θ parameters allows for extra-Poisson variation which can account for other factors that affect the observed data, such as selection or variation in mutation rates.

We assume that α is common across segments, but that each segment has its own β parameter. Thus, by integrating out θ_k we have that

$$p(y_k|\beta) = \frac{\Gamma(\alpha + y_k)\beta^\alpha}{\Gamma(\alpha)y_k!(\beta + 1)^{\alpha+y_k}}.$$

The conjugate prior for β is

$$\pi_0(\beta; a, b) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} \frac{\beta^{a-1}}{(1 + \beta)^{a+b}},$$

which depends on hyperparameters a and b, and is a scaled $F(2a, 2b)$ random variable. If we ignore the $\Gamma(\alpha + y_k)/(\Gamma(\alpha)y_k!)$ in the likelihood, we get that for such a prior (14) becomes

$$P_0(j, i; a, b) = \frac{\Gamma(a + b)\Gamma(a + (i - j)\alpha)\Gamma(b + S)}{\Gamma(a)\Gamma(b)\Gamma(a + b + (i - j)\alpha + S)},$$

where $S = \sum_{k=j+1}^{i} y_k$, is the sum of observations in the segment.

Didelot et al. (2007) show that the distribution of divergences across different genes is bimodal, and give scenarios underwhich such a bimodal distribution could arise. Therefore we use a two-component prior

$$\pi(\beta) = p\pi_0(\beta; a_1, b_1) + (1 - p)\pi_0(\beta; a_2, b_2).$$

This gives us that

$$P(j, i) = pP_0(j, i; a_1, b_1) + (1 - p)P_0(j, i; a_2, b_2).$$

Our model is completed through a geometric distribution for the segment lengths $g(\cdot)$.

We analysed this model using the Forward-Backward algorithm above. Solving the filtering recursions, with $n = 6062$ for one set of hyperparameter values takes about 1 minute on a desktop PC (though in the next Section we show how this CPU cost can be dramatically reduced). We fixed $\alpha = 5$ and estimated the remaining hyperparameters via Monte Carlo EM, which took around 20 iterations to converge. Results are shown in Figure 4. The top plot shows the distribution of mean divergence values α/β across the inferred segments. We can see a clear bi-modality, with modes close to 0.2% and 1%, as noted in Didelot et al. (2007). This may be because of recent gene exchange between the two strains (see Didelot et al., 2007, for fuller discussion of this).

4 Sequential Importance Sampling

We now consider general filtering models for which the filtering recursion (12) cannot be solved analytically. A popular approach in these cases is to use sequential Monte Carlo (SMC), also known as particle filters, to get approximate solutions to the filtering densities (see Liu and Chen, 1998; Doucet et al., 2000, 2001, for more extended reviews of these methods); our exposition here is based on that in Fearnhead (2005b).

The idea of SMC is to approximate the filtering density at time i by a set of particles $\{x_i^{(j)}\}_{j=1}^N$ and associated weights $\{w_i^{(j)}\}_{j=1}^N$ which sum to one. The filtering distribution is approximated by a discrete probability mass function whose support is the set of particles, and which has probability $w_i^{(j)}$ assigned to the jth particle value.

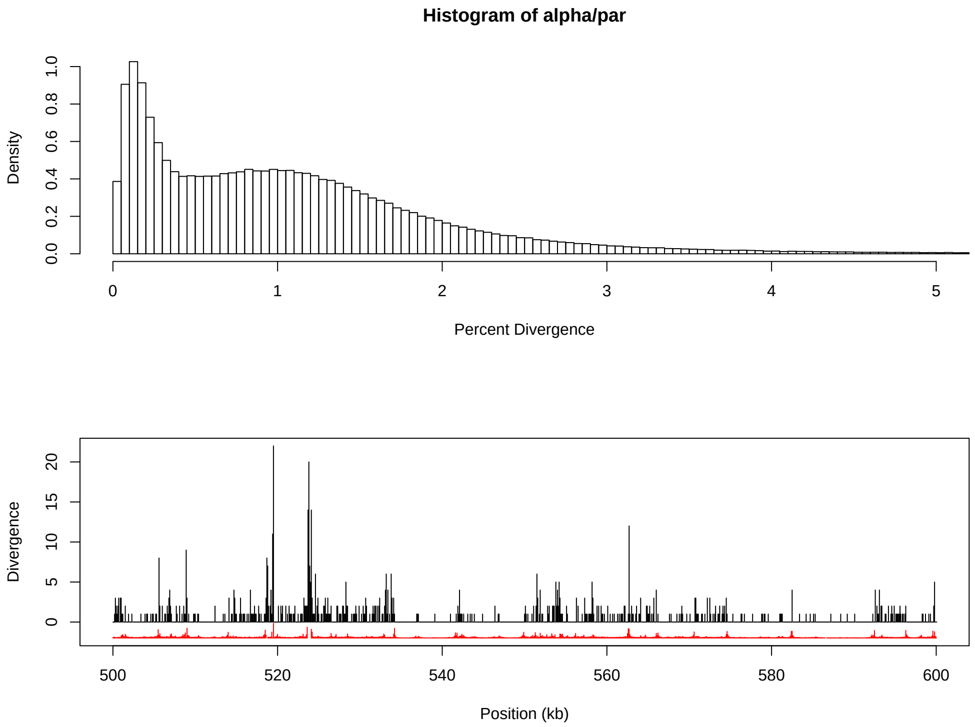

Figure 4: Results of the analysis of the divergence data between *Salmonella* Typhi and *Salmonella* Paratyphi A. (Top) The posterior distribution of the mean divergence α/β across the inferred segments. (Bottom) Data from a 100kb subregion of the 606.2kb region analysed; below the axis we show the posterior probability of a changepoint.

By substituting this particle approximation at time i into (12) we get an approximation to the filtering distribution at time $i+1$, which we denote as $\hat{\pi}_{i+1}(x_{i+1})$, where

$$\hat{\pi}_{i+1}(x_{i+1}) \propto \sum_{j=1}^{N} w_i^{(j)} p(x_{i+1}|x_i^{(j)}) p(y_{i+1}|x_{i+1}). \tag{15}$$

At its simplest, the particle filter can be viewed as the following:

1 **Initiation** Produce a particle approximation to the prior $p(x_1)$.

2 **Iteration** (Time $i+1$.) Given a particle approximation to the filtering distribution at time i, calculate a new particle approximation to $\hat{\pi}_{i+1}(x_{i+1})$.

The key to an efficient algorithm is the method for generating the particle approximation to $\hat{\pi}_{i+1}(x_{i+1})$. Various approaches to this iteration have been proposed. In general the iteration is split into propagation, reweighting and resampling steps. The propagation step generates the particles at time $i+1$, and the reweighting step calculates the particles' weights. The resampling step is optional, and produces a set of equally weighted particles, some of which will be duplicated, which can be viewed as an approximate sample from the posterior.

The propagation and reweighting steps are based on IS. The simplest IS approach, used in Gordon et al. (1993), is to simulate particles from the proposal density

$$q(x_{i+1}) = \sum_{j=1}^{N} w_i^{(j)} p(x_{i+1}|x_i^{(j)}), \tag{16}$$

in the propagation step, and then assign a weight proportional to the likelihood of the resulting particles. This approach performs well if the likelihood is not too peaked compared with the proposal density; but otherwise can produce highly variable weights.

A general framework for implementing SMC, which allows for the design of good IS proposal density, is given by the ASIR filter of Pitt and Shephard (1999). This filter allows for flexibility in the choice of proposal density, and the proposal density can be chosen to take account of the model and the information in the observation at the next time step. In the ASIR filter the proposal is of the form

$$q(x_{i+1}) = \sum_{j=1}^{N} \beta_j q(x_{i+1}|x_i^{(j)}). \tag{17}$$

To simulate from this proposal we first simulate the component, k, from the discrete distribution which assigns probability β_j to value j, and then simulate x_{i+1} from $q(x_{i+1}|x_i^{(j)})$. For a simulated pair (k, x_{i+1}), the new particle is x_{i+1}, and its weight is proportional to

$$\frac{w_i^{(k)} p(x_{i+1}|x_i^{(k)}) p(y_{i+1}|x_{i+1})}{\beta_k q(x_{i+1}|x_i^{(k)})}.$$

In practice $q(x_{i+1})$ is chosen to be as close as possible to $\hat{\pi}_{i+1}(x_{i+1})$. For some problems it is possible to choose $q(x_{i+1}) = \hat{\pi}_{i+1}(x_{i+1})$, whereas for others, approximations of $\hat{\pi}_{i+1}(x_{i+1})$, often based on a Taylor expansion, can be used. See Pitt and Shephard (1999) for more details.

Both for the method of Gordon et al. (1993) and the ASIR filter, the normalising constant of the importance sampling weights at one iteration can be used to estimate the likelihood $p(y_{i+1}|y_i)$ (Kitagawa, 1996). The results estimator of the likelihood can be shown to be unbiased under quite general implementations of the algorithm (Del Moral, 2004; Del Moral and Doucet, 2004).

Any resampling step increases the Monte Carlo variation of the filter. The reason for using resampling steps is linked to early particle filter algorithms (e.g. Gordon et al., 1993; Kong et al., 1994; Liu and Chen, 1995) which only proposed one future particle for each existing particle (that is they sampled a single value from each $p(x_{i+1}|x_i^{(j)})$ at the propagation step). For such algorithms, resampling allows multiple particles to be generated in areas of high posterior probability, which can then independently explore the future of the state. There is still a trade-off between this advantage, and the disadvantage of extra Monte Carlo variation, and the ESS of the weights is often used to guide whether and when to resample (Liu and Chen, 1995). Furthermore there are numerous algorithms for performing resampling while introducing little Monte Carlo variation (Kitagawa, 1996; Liu and Chen, 1998; Carpenter et al., 1999).

It should be noted that for the ASIR framework, resampling naturally occurs within the propagation step. For example the filter of Gordon et al. (1993), which resamples particles independently and then propagates each resampled particle once, is equivalent to the ASIR filter with proposal density (16), while the algorithm of Kong et al. (1994) is equivalent to the ASIR filter with stratified sampling from the proposal density,

$$q(x_{t+1}) = \sum_{i=1}^{N} \frac{1}{N} p(x_{t+1}|x_t^{(i)}).$$

The advantage of viewing the resampling stage as occurring within the propagation stage is that it is easier to relate the stage to its reason, namely that of producing a set of particles, evenly spaced out in areas of high posterior probability, at the next time point. This can help with the choice of when and how to perform resampling, and link this choice with the method for propagation and the specific model of interest. It can also help avoid unnecessary resampling steps, which may occur through the idea of having resampling steps at the end of each iteration of the particle filter. For example, the initial ASIR algorithm of Pitt and Shephard (1999) included an unnecessary resampling algorithm, which can severly reduce the accuracy of the filter: Carpenter et al. (1999) give an example where this unnecessary resampling step reduced the accuracy of the filter by a factor of 2.

There have been further suggestions for improving the efficiency of SMC methods, which include the use of MCMC (Gilks and Berzuini, 2001; Fearnhead, 2002b), and he use of quasi-Monte Carlo methods (Fearnhead, 2005b; L'Ecuyer et al., 2007). A further idea is that of marginalisation (Liu and Chen, 1998; Andrieu and Doucet, 2002) which we demonstrate below on a specific example.

We have described how SMC can be used to solve the filtering recursions. For details of extensions of SMC which are efficient for simulating from $p(\mathbf{x}|\mathbf{y}_{1:n})$ see Kitagawa (1996); Hurzeler and Kunsch (1998); Doucet et al. (2000) and Godsill et al. (2004).

We now consider one example application of the particle filter, which is based on analysis of mixture models, and uses ideas from Fearnhead (2004) (see also Chopin, 2007). This example has specific structure, namely a discreteness of the underlying state, that we can use to design an efficient SMC algorithm. For examples of the application of SMC methods more generally see Liu and Chen (1998); Doucet et al. (2000); Del Moral et al. (2006) and Del Moral et al. (2007). We then revisit Examples 3 and 4, to look at how the resampling idea from SMC can be applied to the problems studied in the earlier sections.

Example 5: Inferring Population Structure from genetic data

Consider genetic data from a set of diploid individuals. We assume that the data consists of the genotype of the individual at L unlinked loci. Thus for each locus we have details of the alleles that are present on each of the two copies of the individual's genome. We further assume that the individuals come from an unknown number of populations, and that there is random-mating within populations. Our assumption of unlinked loci will mean that, conditional on the population of an individual, data at different loci are independent of each other.

We model the data using the no-admixture model of Pritchard et al. (2000a). This model, and its extensions (see for example Pritchard et al., 2000a; Nicholson et al., 2002; Falush et al., 2003), have been very popular for analysing population genetic data; partly because of the importance of detecting and correcting for population structure when performing tests of genetic association (Pritchard et al., 2000b).

Assume that at locus l we have K_l alleles. For population j, denote the frequencies of these alleles as $\mathbf{p}^{(j,l)} = p_1^{(j,l)}, \dots, p_{k_l}^{(j,l)}$. Consider an individual with genotype $\mathbf{y}_i = \{y_{i,l}^{(1)}, y_{i,l}^{(2)}\}_{l=1}^{L}$. Let z_i denote the population from which individual i is from. The conditional likelihood of this data, given $z_i = j$, is

$$p(\mathbf{y}_i|z_i = j) = \prod_{l=1}^{L} p_{y_{i,l}^{(1)}}^{(j,l)} p_{y_{i,l}^{(2)}}^{(j,l)}.$$

Conditional on the assignment of individuals to populations, the likelihood for each individual is independent of each other. We further assume a set of probabilities q_j such that $p(z_i = j) = q_j$.

Our model is finalised through independent Dirichlet priors on $\mathbf{p}^{(j,l)}$ for all j and l; and through a prior on the number of populations and the probabilities $q_j = \Pr(z_i = j)$. Pritchard et al. (2000a) fix the number of populations, M, and assume a dirichlet prior on the $(q_1, \dots, q_M)$. They then approximate the marginal likelihood for M to make inference for the number of populations. They comment on the difficulty of directly inferring M.

We take an alternative approach, and use a mixture Dirichlet process (MDP) model (Ferguson, 1973). Let α be the parameter of this MDP model. One way of viewing such a model is that it is the model of Pritchard et al. (2000a) where the parameters of the Dirichlet prior on $(q_1, \ldots, q_M)$ are $(\alpha/M, \ldots, \alpha/M)$, and we take the limit $M \to \infty$. Note that of interest now is not the number of underlying populations, but the number of populations that are represented by the sample.

We will use the following recursive representation of the MDP model (Blackwell and MacQueen, 1973). Let $\mathbf{z}_{1:i} = (z_1, \ldots, z_i)$ be the population of origin of the first i individuals, and define $m(\mathbf{z}_{1:i})$ to be the number of populations present in $\mathbf{z}_{1:i}$. We number these populations $1, \ldots, m(\mathbf{z}_{1:i})$, and let $n_j(\mathbf{z}_{1:i})$ be the number of the individuals assigned to population j. Then

$$p(z_{i+1} = j|\mathbf{z}_{1:i}) = \begin{cases} n_j(\mathbf{z}_{1:i})/(i+\alpha) & \text{if } j \leq m(\mathbf{z}_{1:i}), \\ \alpha/(i+\alpha) & \text{if } j = m(\mathbf{z}_{1:i}) + 1. \end{cases}$$

The simplest implementation of SMC to this model is to let the state be $\mathbf{z}_{1:i}$ together with the allele frequencies at each locus for each of the $m(\mathbf{z}_{1:i})$ populations. However such an implementation would be impracticable due to the high-dimension of the state. (For example, for the data of Rosenberg et al. (2002) there are 377 loci with multiple alleles at each locus.) However we can use the idea of marginalisation (Liu and Chen, 1998; Chen and Liu, 2000a) to avoid this problem. The idea of marginalisation is to note that conditional on $\mathbf{z}_{1:i}$ we can integrate out the population allele frequencies. Thus we can calculate

$$p(\mathbf{y}_{i+1}, z_i = j|\mathbf{z}_{1:i}, \mathbf{y}_{1:i}) = p(z_{i+1} = j|\mathbf{z}_{1:i}) \prod_{l=1}^{L} p_l(\mathbf{y}_{i+1}|\mathbf{z}_{1:i}, z_{i+1} = j, \mathbf{y}_{1:i}),$$

where $p_l(\mathbf{y}_{i+1}|\mathbf{z}_{1:i}, z_{i+1} = j, \mathbf{y}_{1:i})$ is the conditional probability of the data just at the lth locus. This probability is just a Dirichlet integral, and for brevity we omit details of its calculation.

We implement SMC with the state being $\mathbf{x}_i = \mathbf{z}_{1:i}$. The conditional distribution $p(\mathbf{x}_{i+1}|\mathbf{x}_i)$ is non-zero only for $\mathbf{x}_{i+1}$ and $\mathbf{x}_i$ that agree on allocations of the first i individuals to populations. Thus, if $\mathbf{x}_{i+1} = (\mathbf{x}_i, j)$ then $p(\mathbf{x}_{i+1}|\mathbf{x}_i) = p(z_{i+1} = j|\mathbf{x}_i)$. Note that this distribution takes only a small number, $m(\mathbf{x}_i) + 1$, of possible values; and thus given a set of particles at time i, the approximation of the filtering distribution at time $i+1$, $\hat{\pi}_{i+1}(\mathbf{x}_{i+1})$ defined in (15), can be calculated exactly. If there are N particles at time i, the number of terms in $\hat{\pi}_{i+1}(\mathbf{x}_{i+1})$ will be at least $2N$, thus to avoid an exponentially increasing number of particles, we will need some mechanism for approximating this distribution with N particles.

A simple, and seemingly optimal, approach is to sample N particles from $\hat{\pi}_{i+1}(\mathbf{x}_{i+1})$ – this corresponds to IS with the optimal proposal distribution. However, the discrete nature of the support for $\hat{\pi}_{i+1}(\mathbf{x}_{i+1})$ means that there is a simple way to improve on this. Remember that the aim of SMC is to construct an accurate particle approximation to (15). Now in terms of approximating $\hat{\pi}_{i+1}(\mathbf{x}_{i+1})$, there is no advantage in having multiple copies of the same particle – it will be better to have at most one copy of each particle, but allow the particles to carry different weights. There are various approaches to doing this, and Fearnhead and Clifford (2003) suggest one specific approach that they show satisfies an optimality condition. If we assume that $\hat{\pi}_{i+1}(\mathbf{x}_{i+1})$ has M terms, and the weight assigned to the kth term is $w_{(k)}$, then their algorithm is:

Resampling of Fearnhead and Clifford (2003)

(1) Solve $N = \sum_{k=1}^{M} \min(w^k/c, 1)$ for c. Simulate $U \sim \text{Unif}[0, c]$, and set $k = 1$.

(2) If $w^{(k)} > c$ goto (4).

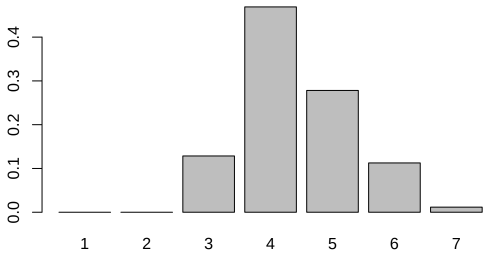

Figure 5: Posterior distribution of the number of populations observed in the sample.

(3) Let $U = U - w^{(k)}$. If $U > 0$ then let $w^{(k)} = 0$; else let $U = U + c$, and $w^{(k)} = c$.

(4) Let $k = k + 1$. If $k \leq M$ goto (2); else end.

The particles with zero weight are removed. The choice of c in (1) ensures that there will be N particles with non-zero weights.

To show the advantage of this approach over simulating from $\hat{\pi}_{i+1}(\mathbf{x}_{i+1})$ we analysed a subset of the data from Jorde et al. (1995). For simplicity we analysed a sample of 39 genotypes, sampled from 20 Europeans (all French) and 19 Africans (all Sotho). For each individual we have data from 30 biallelic loci. We analysed the data with a uniform prior on population allele frequencies at each locus, and the parameter of DPM was $\alpha = 0.1$. The latter parameter governs the prior on the number of populations observed in a sample; and the mean of this is approximately $\alpha \log(n)$, for a sample of size n. Our prior thus penalises large number of populations. We ran the SMC method with $N = 10,000$ particles for both (i) sampling particles from $\hat{\pi}_{i+1}(\mathbf{x}_{i+1})$ using the stratified sampling algorithm of Carpenter et al. (1999); and (ii) using the resampling algorithm described above. To compare the methods we ran each method 100 independent times on the data; and looked at the variance of the estimates of the log marginal likelihood. The variance was 1.1 and 0.081 for (i) and (ii) respectively; which corresponds to a 13-fold reduction in the variance.

The posterior distribution for the number of populations observed in the sample, obtained from a single run of the SMC method, is shown in Figure 5. This posterior gives no mass to 2 populations, which was the number of geographic populations the data was sampled from, and has a modal estimate of 4. The posterior distribution appears to group all the Sotho individuals in a single population, but then allows for 2 or more populations for the French individuals. To check whether this is an artefact of any problems with the SMC algorithm, we calculated the conditional likelihood of the data given (i) assignment of individuals to their geographic population of origin; (ii) assignment of individuals given by the particle with largest weight, which had 4 populations. The log-likelihood of the latter was 5.54 greater than the log-likelihood of the former; which suggests that the model does indeed prefer more than 2 populations for the data.

As discussed by Pritchard et al. (2000a), inference for the number of populations can be sensitive to the priors chosen for the population allele frequencies; though it should be noted that we chose our priors to penalise too many populations. Alternative choices of priors we have tested have produced posteriors which, if anything, tend to suggest more populations within the sample.

Example 4 Revisited

We now look at how resampling ideas can be used to improve the Forward-Backward algorithm for changepoint models that was presented in Section 3. Firstly, it is easy to see the link between this Forward-Backward algorithm and the SMC algorithm described above. Within the Forward-Backward algorithm, at any time i the filtering distribution $p(x_i|\mathbf{y}_{1:i})$ is a discrete distribution which can take i values. Thus it can be described exactly by a set of i particles, taking the values $0, 1, \ldots, i-1$, and corresponding weight (i.e. probability). One problem with the Forward-Backward algorithm is that the number of particles needed to describe the filtering distribution increases linearly with i.

Fearnhead and Liu (2007) suggest approximating the filtering density at time i with fewer than i particles. This will introduce approximation error, but with the gain of a reduction in computational cost, which will now be linear in the sample size. One approach to doing this is through resampling. One efficient resampling algorithm, called stratified rejection control, is the same as the resampling algorithm of Fearnhead and Clifford (2003), but with step (1) replaced by

(1') Assume we have a set of *ordered* particles $x^{(1)}, \ldots, x^{(M)}$, with corresponding normalised weights $w^{(1)}, \ldots, w^{(M)}$. Fix a constant $c < 1$, and simulate $U \sim \text{Unif}[0, c]$. Set $k = 1$.

This algorithm outputs a new set of weights; the subset of particles with weight 0 can be removed. The idea is that particles with large weights (as defined by the threshold c) are kept without resampling. Resampling is applied to the remaining particles. However due to ordering in step (1), this resampling occurs in a stratified manner such that if a particle is resampled then nearby particles are less likely to be resampled.

The algorithm is similar to that describe in Example 5, except for the ordering of particles and the definiton of c. This algorithm is also closely related to the rejection control idea of Liu et al. (1998). The only difference is the ordering of particles and the stratified sampling. However, this stratified resampling has the nice property that the error introduced by the resampling, as measured by the Kolmogorov-Smirnov statistic, is bounded above by c. Thus c governs the amount of error introduced by resampling. Note that the number of particles that are kept by this algorithm will naturally vary over time – depending on how easy it is to approximate the filtering distribution at different times.

To show the potential gains of resampling, we implemented the Forward-Backward algorithm of example 4, but with resampling with $c = 10^{-6}$. This reduced the average number of particles of the algorithm by a factor of 50, and the CPU cost of approximating the filtering densities was less than 2 seconds. The effect of the resampling on the posterior distribution was negligible.

IS Revisited

It is also possible to consider applying SMC ideas to the IS methods described in Section 2. Remember that the aim was to estimate a likelihood based on sampling the hidden path, $\mathbf{x}$. The key idea is that rather than simulating $\mathbf{x}$ values one at a time; they can be simulated concurrently, and resampling ideas used. The first time this was suggested was in a comment on the article of Stephens and Donnelly (2000) by Chen and Liu (Chen and Liu, 2000b).

The first thing to note is that while in Section 2 we gave formulae for the IS weight for a complete path; these weights factorise and can be calculated sequentially through time as we simulate each component of $\mathbf{x}$. Thus, given an incomplete path $\mathbf{x}_{1:i}$, it is possible to calculate an IS weight to associate with that path. For Example 1, this IS weight would be of the form

$$\prod_{k=1}^{i} \left(\frac{\pi(x_{kh}|x_{(k-1)h})}{q(x_{kh}|x_{(k-1)h})} \right),$$

where $\pi(\cdot|\cdot)$ denotes the Euler approximation to the transition density of the diffusion over time interval h; and $q(\cdot|\cdot)$ denotes the proposal distribution.

The implementation of this idea involves simulating a batch of x_1 values (henceforth we call these particles, due to the relation to SMC/particle filter algorithms) together with their associated IS weights; and then from each x_1 particle simulate a particle for $\mathbf{x}_{1:2}$ and the associated IS weight; and to repeat this recursively over time. Assume that at time i we have sampled N particles for $\mathbf{x}_{1:i}$ from our proposal. Denote these values by $\mathbf{x}_{1:i}^{(j)}$, for $j = 1, \dots, N$; and let $w_i^{(j)}$ denote the IS weight associated with $\mathbf{x}_{1:i}^{(j)}$. Due to the Markov property of the models, we need not store the whole path of each particle, just its current value $\mathbf{x}_i^{(j)}$, as the final estimate of the likelihood depends only on the current state of the particles and their associated weight.

If we implemented this as described, the results would be no different than standard IS – the only difference is in the order in which we have done the simulation. However Chen and Liu (2000b) show that using resampling can improve the performance of the resulting estimate. The idea is that if at time i the weights are sufficiently skewed (e.g. in terms of having a low ESS), we can perform resampling to produce a set of equally weighted particles; and then propagate these forward. Hopefully this resampling will produce multiple copies of the "good" particles, and these multiple copies will be able to independently explore the future of the $\mathbf{x}$ path conditional on $\mathbf{x}_{1:i}$.

There are two difficulties with this idea, the first is that in some applications (e.g. Example 3), the length of the $\mathbf{x}$ paths can be random. This makes it less clear whether it is fair to compare IS weights at fixed time-points i – as some particles will have almost completed paths, and others may still need to be extended over a considerable number of time steps. The second problem, which is related and perhaps more fundamental, is that if we are to perform resampling, then we want to resample particles with probabilities proportional to their expected final weight. In some situations this may not be highly correlated with the current weight of the particle, and resampling will actually increase the variance of the IS estimator. This particularly occurs if we have been able to use the information in the data to construct a good IS proposal distribution. In these cases our proposal distribution may specfically simulate paths that initially have low IS weights (relative to other paths up to the same time point) because it expects the relative IS weight to increase as we simulate more of the path. Resampling has the potential problem of removing such paths.

To give a trivial example, consider the model in Example 3, but with data collected at a single time point. As pointed out, in this case we can implement an exact IS method to calculate the likelihood or simulate from the missing data. We implemented such a method for simulated data with $\theta = 10$ and a sample size of 200. We simulated $10,000$ paths from the optimal proposal, and in Figure 6 we show the values of the (normalised) weights of the particles through time. As the IS proposal is optimal, all final IS weights are equal to the likelihood; however there is substantial variation in the weights through time. For example at time 150, the ESS of the weights is 26. However if we implement resampling at this time-point we will end up with final IS weights that are highly variable: one implementation of this produced final IS weight with ESS of 22.

This is an extreme example, as in this case there is no correlation between a path's final IS weight and its weight at an earlier time i. More generally we would expect there to be correlation between these two weights, but this could be substantially less than 1; and therefore a simple implementation of resampling may actually increase the variance of the IS weights.

Chen and Liu (2000b) and Chen et al. (2005) suggest a very clever way around these problems. Their idea is to introduce stopping times; to simulate particles forward until these stopping times; and only consider resampling among particles at the same stopping time. The idea is

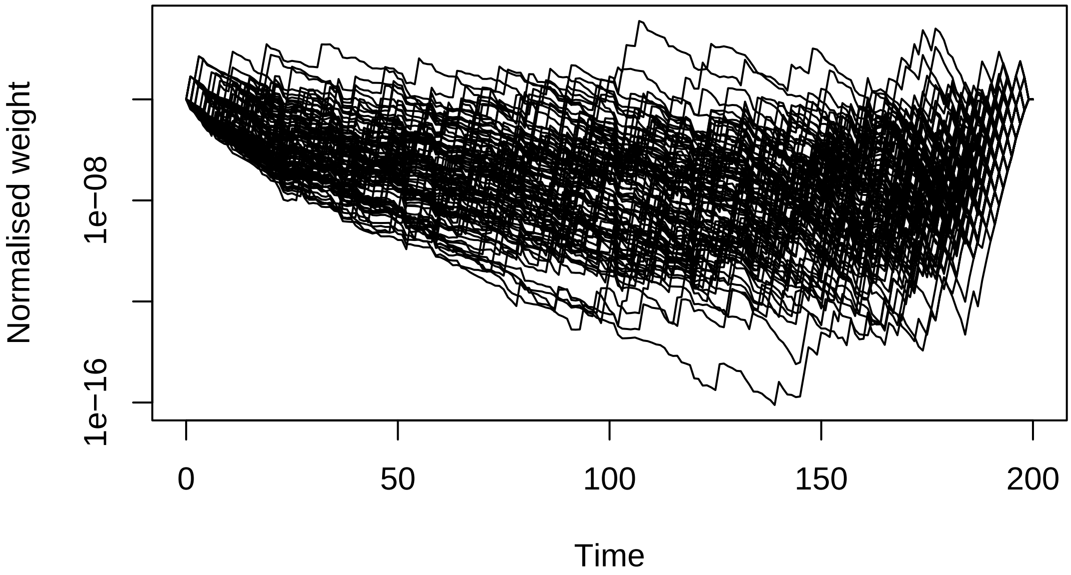

Figure 6: Normalised IS weights versus time for 100 (out of 10000) pahts. The model is described in Example 3, but we have assumed data collected at a single time-point.

that if a stopping time is chosen appropriately, the weights at that stopping time will be good predictors of the expected final weights of the particles. In Chen et al. (2005) there are a number of examples of how this method can work well in practice.

There is a further approach that may be possible for this problem. This is to try and estimate the expected final IS weight for the particles, and resample based on that. In Example 1, the expected final IS weight for a particle $\mathbf{x}_{1:i}$ with weight w_i will be $w_i p_\Delta(\mathbf{y}|\mathbf{x}_i)$, where $p_\Delta(\cdot|\cdot)$ is the transition density of the diffusion over a time interval $\Delta = T - ih$. This could be estimated via the Euler approximation.

We tried this approach. We analysed the same CIR diffusion as in Example 1, with $X_0 = 4.9$, $X_T = 5.0$, $T = 0.2$ and $h = 0.01$, using the IS method of Pederson. We resampled based on $w_i^* = w_i \pi_\Delta(\mathbf{y}|\mathbf{x}_i)$, where we approximate $\pi_\Delta(\mathbf{y}|\mathbf{x}_i)$ via the Euler approximation. We compared no resampling, together with resampling when the ESS of the w_i^*s was less than $N/2$. Our comparison was based on the variance of the estimate of the transition probability across 100 independent runs of the two methods. Using $N = 10,000$, we found the variance of the estimates when no resampling was used was 1.4×10^{-4}, while using resampling reduced the variance to 9.0×10^{-5}. The reduction in variance was robust to the choice of threshold. However, the efficiency of this method depends on the accuracy of the approximation to $\pi_\Delta(\mathbf{y}|\mathbf{x}_i)$, and it becomes less efficient for larger T; and for sufficiently large T resampling actually reduces the accuracy of the IS approach.

5 Discussion

We have described a number of related approaches for analysing complex stochastic systems without resort to MCMC. The first of these was based on IS. The key to obtaining an efficient

IS algorithm is to design a good proposal distribution. We have shown the form of the optimal proposal distribution, and given three examples of how to use this to construct a good proposal distribution in practice. A common strategy to Examples 1 and 3 was to consider the form of the optimal proposal distribution when the further time to the observation, Δ is both large and small. The optimal proposal can be calculated in the limits as $\Delta \to \infty$ and $\Delta \to 0$, and these limiting results can help guide the choice of proposal for intermediate values of Δ.

The second approach was the Forward-Backward algorithm. While the application of the Forward-Backward algorithm to hidden Markov models is both well-known, and used within Bayesian analysis, we have highlighted some recent work which enables the Forward-Backward algorithm to be applied to other models, in particular changepoint models. As can be seen by our Example 4, such an approach offers a simple and efficient alternative to reversible jump MCMC methods, and has the advantage of allowing iid draws from the posterior. The final approach was SMC, or particle filter algorithms. These have become very popular in recent years, particularly for online problems, for which MCMC methods are not suitable. However, SMC methods can be competitive even for batch analysis of data (such as Example 5). This was first suggested by Chopin (2002) (see also Ridgeway and Madigan, 2003). A general framework for implementing SMC methods for batch problems is described in Del Moral et al. (2006); and both this paper and Del Moral et al. (2007) contain examples of the situations where SMC methods are more efficient than MCMC. For an alternative approach see Cappe et al. (2004); Celeux et al. (2006), and see Jasra et al. (2007) for a comparison of methods with population MCMC.

We have also shown how resampling ideas from SMC can be applied to both the Forward-Backward algorithm and the IS approach. The resampling step with SMC is fundamental to the good theoretical properties of the method, particularly for large data sets (Del Moral and Guionnet, 2001; Künsch, 2005); as such being able to apply these ideas to the IS methods discussed earlier has the potential at least to lead to large improvements in efficiency. However, it is not currently clear how to implement resampling ideas for these models in any generality.

The methods we have been looking at produce estimates of the likelihood, and simulate from the conditional distribution of the hidden data, for specified values of parameters. An open question, is how best to implement these methods when the parameters are unknown. There has been research in this area for SMC methods; in particular the use of Kernel Density estimation (Liu and West, 2001) and the algorithm of Storvik (2002) (see also the related idea of Fearnhead, 2002b). However even these methods are known to struggle for large data sets. For IS methods, there has been work on producing smooth estimates of the likelihood curve (e.g. Beskos et al., 2007); while the Forward-Backward algorithm has been used within MCMC methods (Scott, 2002). Given the gain in computation that is possible by using resampling ideas with the Forward-Backward algorithm (see Example 4), it would be good to be able to use such approximate methods such as SMC within MCMC. Some ideas along this line can be found in Neal (2003).

Acknowledgements I thank Xavier Didelot for providing the data used in Example 4. This work was supported by EPSRC grant GR/T19698.

References

Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society, Series B*, 64:827–836.

Bahlo, M. and Griffiths, R. C. (1998). Inference from gene trees in a subdivided population.

Theoretical Population Biology, 57:79–95.

Barry, D. and Hartigan, J. A. (1992). Product partition models for change point problems. *The Annals of Statistics*, 20:260–279.

Barry, D. and Hartigan, J. A. (1993). A Bayesian analysis for change point problems. *Journal of the American Statistical Society*, 88:309–319.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximisation technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stats.*, 41:164–171.

Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2007). Monte Carlo maximum likelihood estimation for discretely observed diffusion processes. *Annals of Statistics*.

Beskos, A., Papaspiliopoulos, O., Roberts, G. O., and Fearnhead, P. (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society Series B*, 68:333–382.

Bhattacharya, S., Gelfand, A. E., and Holsinger, K. E. (2007). Model fitting and inference under latent equilibrium processes. *Statistics and Computing*, 17:193–208.

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Polya urn schemes. *Annals of Statistics*, 1:353–355.

Boys, R. J., Wilkinson, D. J., and Kirkwood, T. B. L. (2007). Bayesian inference for a discretely observed stochastic kinetic model. *Submitted*.

Cappe, O., Guillin, A., Marin, J. M., and Robert, C. P. (2004). Population monte carlo. *Journal of Computational and Graphical Statistics (to appear)*, 13:907–929.

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.

Celeux, G., Marin, J., and Robert, C. P. (2006). Iterated importance sampling in missing data problems. *Computational Statistics and Data Analysis*, 50:3386–3404.

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

Chen, Y. and Liu, J. S. (2000b). Comment on 'Inference in molecular population genetics' by M. Stephens and P. Donnelly. *Journal of the Royal Statistical Society: series B*, 62:644–645.

Chen, Y., Xie, J., and Liu, J. S. (2005). Stopping-time resampling for Monte Carlo methods. *Journal of the Royal Statistical Society Series B*, 67:199–217.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89:539–551.

Chopin, N. (2007). Inference and model choice for time-ordered hidden markov models. *Journal of the Royal Statistical Society, Series B*, 69:269–284.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer, New York.

Cox, J. C., Ingersoll, Jr, J. E., and Ross, S. A. (1985). A theory of the term structure of interest rates. *Econometrica*, 53:385–407.

De Iorio, M. and Griffiths, R. C. (2004a). Importance sampling on coalescent histories. I. *Advances in Applied Probability*, 36:417–433.

De Iorio, M. and Griffiths, R. C. (2004b). Importance sampling on coalescent histories. II: Subdivided population models. *Advances in Applied Probability*, 36:434–454.

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems With Applications.* Springer, New York.

Del Moral, P. and Doucet, A. (2004). Particle motions in absorbing medium with hard and soft obstacles. *Stochastics Analysis and Applications*, 22:1175–1207.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, 68:411–436.

Del Moral, P., Doucet, A., and Jasra, A. (2007). Sequential Monte Carlo for Bayesian computation. In Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M., and West, M., editors, *Bayesian Statistics 8*, pages 115–148, Oxford. Oxford University Press.

Del Moral, P. and Guionnet, A. (2001). On the stability of interactin processes with applications to filtering and genetic algorithms. *Ann. Inst. of H. Poincaré Probab. Statist.*

Delyon, B. and Hu, Y. (2006). Simulation of conditioned diffusion and application to parameter estimation. *Stochastic Processes and their Applications*, 116:1660–1675.

Didelot, X., Achtman, M., Parkhill, J., Thomson, N. R., and Falush, D. (2007). A bimodal pattern of relatedness between the *salmonella* Paratyphi A and Typhi genomes: Convergence or divergence by homologous recombination? *Genome Research*, 17:61–68.

Donnelly, P. and Kurtz, T. (1996). A countable representation of the Fleming-Viot measure-valued diffusion. *The Annals of Probability*, 24:698–742.

Donnelly, P. and Tavaré, S. (1995). Coalescents and genealogical structure under neutrality. *Annual Review of Genetics*, 29:401–421.

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

Doucet, A., Godsill, S. J., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208.

Drummond, A. J., Nicholls, G. K., Rodrigo, A. G., and Solomon, W. (2002). Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. *Genetics*, 161:1307–1320.

Drummond, A. J., Rambaut, A., Shapiro, B., and Pybus, O. G. (2005). Bayesian coalescent inference of past population dynamics from molecular sequences. *Molecular Biology and Evolution*, 22:1185–1192.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press.

Durham, G. B. and Gallant, A. R. (2002). Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *Journal of Business and Economic Statistics*, 20:297–338.

Falush, D., Stephens, M., and Pritchard, J. K. (2003). Inference of population structure using multilocus genotype data: Linked loci and correlated allele frequencies. *Genetics*, 164:1567–1587.

Falush, D., Torpdahl, M., Didelot, X., Conrad, D. F., Wilson, D. J., and Achtman, M. (2006). Mismatch induced speciation in Salmonella: model and data. *Philosophical Transactions of the Royal Society of London, series B*, 361:2045–2053.

Fearnhead, P. (2002a). The common ancestor at a non-neutral locus. *Journal of Applied Probability*, 39:38–54.

Fearnhead, P. (2002b). MCMC, sufficient statistics and particle filters. *Journal of Computational and Graphical Statistics*, 11:848–862.

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

Fearnhead, P. (2005a). Exact Bayesian curve fitting and signal segmentation. *IEEE Transactions on Signal Processing*, 53:2160–2166.

Fearnhead, P. (2005b). Using random Quasi-Monte Carlo within particle filters, with application to financial time series. *Journal of Computational and Graphical Statistics*, 14:751–769.

Fearnhead, P. (2006). Exact and efficient inference for multiple changepoint problems. *Statistics and Computing*, 16:203–213.

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 Donnelly, P. (2001). Estimating recombination rates from population genetic data. *Genetics*, 159:1299–1318.

Fearnhead, P. and Donnelly, P. (2002). Approximate likelihood methods for estimating local recombination rates (with discussion). *Journal of the Royal Statistical Society, series B*, 64:657–680.

Fearnhead, P. and Liu, Z. (2007). Online inference for multiple changepoint problems. *To appear in Journal of the Royal Statistical Society Series B.*

Fearnhead, P. and Meligkotsidou, L. (2004). Exact filtering for partially-observed continuous-time Markov models. *Journal of the Royal Statistical Society, series B*, 66:771–789.

Fearnhead, P., Papaspiliopoulos, O., and Roberts, G. O. (2007). Particle filters for partially-observed diffusions. *Submitted to Journal of the Royal Statistical Society Series B.*

Fearnhead, P. and Sherlock, C. (2006). Bayesian analysis of Markov modulated Poisson processes. *Journal of the Royal Statistical Society, Series B*, 68:767–784.

Fearnhead, P. and Vasileiou, D. (2007). Bayesian analysis of isochores. *Submitted.* available from `www.maths.lancs.ac.uk/∼fearnhea/publications`.

Felsenstein, J. and Churchill, G. A. (1996). A Hidden Markov Model approach to variation among sites in rate of evolution. *Molecular Biology and Evolution*, 13:93–104.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1:209–230.

Gamerman, D. (2006). *Markov Chain Monte Carlo: Stochastic Simulation For Bayesian Inference*. Taylor and Francis Ltd, UK.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B*, 63:127–146.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte Carlo smoothing for non-linear time series. *Journal of the American Statistical Association*, 99:156–168.

Golightly, A. and Wilkinson, D. J. (2005). Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61:781–788.

Golightly, A. and Wilkinson, D. J. (2006). Bayesian sequential inference for stochastic kinetic biochemical network models. *Journal of Computational Biology*, 13:838–851.

Gordon, N., Salmond, D., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE proceedings-F*, 140:107–113.

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

Griffiths, R. C. and Tavaré, S. (1994). Unrooted genealogical tree probabilities in the infinitely-many-sites model. *Mathematical Biosciences*, 127:77–98.

Hurzeler, M. and Kunsch, H. R. (1998). Monte Carlo approximations for general state-space models. *Journal of Computational and Graphical Statistics*, 7:175–193.

Jasra, A., Stephens, D. A., and Holmes, C. (2007). On population-based simulation for statics inference. *Statistics and Computing*, page To appear.

Jorde, L. B., Bamshad, M. J., Watkins, W. S., Zenger, R., Fraley, A. E., Krakowiak, P. A., Carpenter, K. D., Soodyall, H., Jenkins, T., and Rogers, A. R. (1995). Origins and affinities of modern humans: a comparison of mitochondrial and nuclear genetic data. *American Journal of Human Genetics*, 57:523–538.

Juang, B. H. and Rabiner, L. R. (1991). Hidden Markov models for speech recognition. *Technometrics*, 33:251–272.

Kalman, R. and Bucy, R. (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering, Transacation ASME series D*, 83:95–108.

Kimura, M. and Crow, J. (1964). The number of alleles that can be maintained in a finite population. *Genetics*, 49:725–738.

Kingman, J. F. C. (1982). The coalescent. *Stochastic Processes and their Applications*, 13:235–248.

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

Kloeden, P. E. and Platen, E. (1992). *Numerical solution of stochastic differential equations*. Springer, New York.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89:278–288.

Kou, S. C., Zhou, Q., and Wong, W. H. (2006). Equi-energy sampler with applications in statistical inference and statistical mechanics. *Annals of Statistics*, 34:1581–1619.

Kschischang, F. R., Frey, B. J., and Loeliger, H. (2001). Factor graphs and the sum-product algorithm. *IEEE transactions on Information Theory*, 47.

Künsch, H. R. (2005). Monte Carlo filters:Algorithms and theoretical analysis. *Annals of Statistics*, 33:1983–2021.

Lander, E. S. and Green, P. (1987). Construction of multilocus genetic linkage maps in humans. *Proceeding of the National Academy of Sciences, USA*, 84:2363–2367.

L'Ecuyer, P., Lécot, C., and Tuffin, B. (2007). A randomized quasi-Monte carlo simulation method for Markov chains. *To appear in Operations Research*.

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation based filtering. In Doucet, A., de Freitas, J. F. G., and Gordon, N. J., editors, *Sequential Monte Carlo in Practice*, pages 197–223, New York. Springer-Verlag.

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. (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90:567–576.

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.

Liu, J. S. and Lawrence, C. E. (1999). Bayesian inference on biopolymer models. *Bioinformatics*, 15:38–52.

Maiden, M. C. J., Bygraves, J. A., Feil, E., Morelli, G., Russell, J. E., Urwin, R., Zhang, Q., Zhou, J., Zurth, K., Caugant, D. A., Feavers, I. M., Achtman, M., and Spratt, B. G. (1998). Multilocus sequence typing: A portable approach to the identification of clones within populations of pathogenic microorganisms. *Proceedings of the National Acadamey of Science, USA*, 95:3140–3145.

Meng, X. and Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, 6:831–860.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equations of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21:1087–1091.

Neal, R. M. (2003). Markov chain sampling for non-linear state space models using embedded hidden Markov models. Available from http://www.cs.toronto.edu/∼radford/emb-hmm.abstract.html.

Nicholson, G., Smith, A. V., Jónsson, F., Gústafsson, O., Stefánsson, K., and Donnelly, P. (2002). Assessing population differentiation and isolation from single-nucleotide polymorphism data. *Journal of the Royal Statistical Society Series B*, 64:695–715.

Papaspilopoulos, O., Roberts, G. O., and Sköld, M. (2003). Non-centred parameterisations for hierarchical models and data augmentation (with discussion). In Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M., and West, M., editors, *Bayesian statistics 7*, London. Clarendon Press.

Pedersen, A. R. (1995). A new approach to maximum likelihood estimation of stochastic differential equations based on discrete observations. *Scandinavian Journal of Statistics*, 22:55–71.

Pitt, M. (2007). Smooth particle filters for likelihood evaluation and maximisation. *Submitted.* Available from `http://www2.warwick.ac.uk/fac/soc/economics/staff/faculty/pitt/publications/`.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94:590–599.

Pritchard, J. K., Stephens, M., and Donnelly, P. (2000a). Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959.

Pritchard, J. K., Stephens, M., Rosenberg, N. A., and Donnelly, P. (2000b). Association mapping in structured populations. *American Journal of Human Genetics*, 67:170–181.

Rabiner, L. R. and Juang, B. H. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15.

Redelings, B. D. and Suchard, M. A. (2005). Joint Bayesian estimation of alignment and phylogeny. *Systematic Biology*, 54:401–418.

Ridgeway, G. and Madigan, D. (2003). A sequential Monte Carlo method for Bayesian analysis of massive datasets. *Data Mining and Knowledge Discovery*, 7:301–319.

Ripley, B. D. (1987). *Stochastic Simulation.* New York: Wiley and Sons.

Roberts, G. O. and Stramer, O. (2001). On inference for partially observed nonlinear diffusion models using the Metropolis-Hastings algorithm. *Biometrika*, 88:603–621.

Rogers, L. C. G. and Williams, D. (2000). *Diffusions, Markov processes and Martingales, Vol. 1.* Cambridge University Press, Cambridge, UK.

Rosenberg, N. A., Pritchard, J. K., Weber, J. L., Cann, H. M., Kidd, K. K., Zhivotovsky, L. A., and Feldman, M. W. (2002). Genetic structure of human populations. *Science*, 298:2381–2385.

Scott, S. L. (1999). Bayesian analysis of a two state Markov modulated Poisson process. *Journal of Computational and Graphical Statistics*, 8:662–670.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97:337–351.

Stephens, D., Jasra, A., and Holmes, C. (2007). On population-based simulation for statistical inference. *Statistics and Computing.* To appear.

Stephens, M. (1999). Problems with computational methods in population genetics. Contribution to the 52nd session of the International Statistical Institute.

Stephens, M. (2000). Times on trees and the age of an allele. *Theoretical Population Biology*, 57:109–119.

Stephens, M. and Donnelly, P. (2000). Inference in molecular population genetics (with discussion). *Journal of the Royal Statistical Society, Series B*, 62:605–655.

Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transaction on Signal Processing*, 50:281–289.

Stramer, O. (2007). On simulated likelihood of discretely observed diffusion processes and comparison to closed form approximation. *To appear in Journal of Computational and Graphical Statistics.*

Stramer, O. and Yan, J. (2007). Asymptotics of an efficient Monte Carlo estimation for the transition density of diffusion processes. *To appear in Methodology and Computing in Applied Probability.*

Wakeley, J. (2007). *Coalescent Theory: An Introduction.* Roberts and Company, Denver, Colorado, USA.

West, M. and Harrison, J. (1989). *Bayesian forecasting and dynamic models.* Springer-Verlag, New York.

Wilkinson, D. J. (2006). *Stochastic Modelling for Systems Biology.* Chapman and Hall/CRC Press, Boca Raton, Florida.

Wilson, D. J. and Fearnhead, P. (2007). Calibrating the rate of evolution of *campylobacter*. In preparation.

Yao, Y. (1984). Estimation of a noisy discrete-time step function: Bayes and empirical Bayes approaches. *The Annals of Statistics*, 12:1434–1447.