Sequential Monte Carlo Samplers

Pierre Del Moral

Université Nice Sophia Antipolis, France

Arnaud Doucet†

University of British Columbia, Canada

Ajay Jasra

University of Oxford, UK

Summary. In this paper, we propose a methodology to sample sequentially from a sequence of probability distributions known up to a normalizing constant and defined on a common space. These probability distributions are approximated by a cloud of weighted random samples which are propagated over time using Sequential Monte Carlo methods. This methodology allows us to derive simple algorithms to make parallel Markov chain Monte Carlo algorithms interact in a principled way, to perform global optimization and sequential Bayesian estimation and to compute ratios of normalizing constants. We illustrate these algorithms for various integration tasks arising in the context of Bayesian inference.

Keywords: Importance Sampling, Ratio of Normalizing Constants, Resampling, Markov chain Monte Carlo, Sequential Monte Carlo, Simulated Annealing.

1. Introduction

Consider a sequence of probability measures $\{\pi_n\}_{n\in\mathbb{T}}$ defined on a common measurable space $(E,\mathcal{E})$, where $\mathbb{T}=\{1,\ldots,p\}$. For ease of presentation, we will assume that each $\pi_n(\mathrm{d}x)$ admits a density $\pi_n(x)$ with respect to a $\sigma-$finite dominating measure denoted $\mathrm{d}x$. We will refer to n as the time index; this variable is simply a counter and need not have any relation with 'real time'. We also denote, by E_n, the support of π_n; that is $E_n=\{x\in E:\pi_n(x)>0\}$. In this paper, we are interested in sampling from the distributions $\{\pi_n\}_{n\in\mathbb{T}}$ *sequentially*; i.e. first sampling from π_1, then from π_2 and so on.

This problem arises in numerous applications. In the context of sequential Bayesian inference, π_n could be the posterior distribution of a parameter given the data collected until time n; e.g. $\pi_n(x)=p(x|y_1,\ldots,y_n)$. In a batch setup where a fixed set of observations $y_1,\ldots,y_p$ is available, one could also consider the sequence of distributions $p(x|y_1,\ldots,y_n)$ for $n\leq p$ for the following two reasons. First, for large datasets, standard simulation methods such as Markov Chain Monte Carlo (MCMC) methods require a complete 'browsing' of the observations, in contrast, a sequential strategy may have reduced computational complexity. Second, by including the observations one at a time, the posterior distributions exhibit a beneficial tempering effect (Chopin, 2002). Alternatively, we may want to move from a tractable (easy to sample) distribution π_1 to a distribution of interest, π_p, through a sequence of artificial intermediate distributions (Neal, 2001). In the context of optimization,

†*Address for correspondence:* Department of Statistics and Department of Computer Science, University of British Columbia, Vancouver, BC, Canada. Email: arnaud@stat.ubc.ca.

First version December 2002, Revised May 2004 and December 2005.

and in a manner similar to simulated annealing, one could also consider the sequence of distributions $\pi_n(x) \propto [\pi(x)]^{\phi_n}$ for an increasing schedule $\{\phi_n\}_{n\in\mathbb{T}}$.

The tools favoured by statisticians, to sample from complex distributions, are MCMC methods (see, for example, Robert and Casella (2004)). To sample from π_n, MCMC methods consist of building an ergodic Markov kernel K_n with invariant distribution π_n using Metropolis-Hastings (MH) steps and Gibbs moves. MCMC algorithms have been successfully applied to many problems in statistics (e.g. mixture modelling (Richardson and Green, 1997) and changepoint analysis (Green, 1995)). However, in general, there are two major drawbacks with MCMC. It is difficult to assess when the Markov chain has reached its stationary regime and it can easily become trapped in local modes. Moreover, MCMC cannot be used in a sequential Bayesian estimation context.

In this paper, we propose a different approach to sample from $\{\pi_n\}_{n\in\mathbb{T}}$ based upon Sequential Monte Carlo (SMC) methods (Del Moral, 2004; Doucet *et al.*, 2001; Liu, 2001). Henceforth, the resulting algorithms will be called SMC samplers. More precisely, this is a complementary approach to MCMC, as MCMC kernels will often be ingredients of the methods proposed. SMC methods have been recently studied and used extensively in the context of sequential Bayesian inference. At a given time n, the basic idea is to obtain a large collection of N weighted random samples $\left\{W_n^{(i)}, X_n^{(i)}\right\}$ ($i = 1, \ldots N$, $W_n^{(i)} > 0$; $\sum_{i=1}^N W_n^{(i)} = 1$) named particles whose empirical distribution converges asymptotically ($N \to \infty$) to π_n; i.e. for any π_n-integrable function $\varphi : E \to \mathbb{R}$

$$\sum_{i=1}^{N} W_n^{(i)} \varphi\left(X_n^{(i)}\right) \xrightarrow{a.s.} \mathbb{E}_{\pi_n}(\varphi)$$

where

$$\mathbb{E}_{\pi_n}(\varphi) = \int_E \varphi(x)\, \pi_n(x)\, \mathrm{d}x. \tag{1}$$

and a.s. denotes almost sure convergence. These particles are carried forward over time using a combination of sequential Importance Sampling (IS) and resampling ideas.

Standard SMC algorithms in the literature do not apply to the problems described above. This is because these algorithms deal with the case where the target distribution of interest, at time n, is defined on S_n with $\dim(S_{n-1}) < \dim(S_n)$; e.g. $S_n = E^n$. Conversely, we are interested in the case where the distributions $\{\pi_n\}_{n\in\mathbb{T}}$ are all defined on a common space E. Our approach has some connections to adaptive IS methods (West, 1993; Oh and Berger, 1993; Givens and Raftery, 1996), Resample-Move (RM) strategies (Chopin, 2002; Gilks and Berzuini, 2001), Annealed IS (AIS) (Neal, 2001) and Population Monte Carlo (Cappé *et al.*, 2004) which are detailed in Section 3. However, the generic framework we present here is more flexible. It allows us to define general moves and can be used in scenarios where previously developed methodologies do not apply (see Section 5). Additionally, we are able to develop new algorithms to make parallel MCMC runs interact in a simple and principled way, to perform global optimization or solve sequential Bayesian estimation problems. It is also possible to estimate ratios of normalizing constants as a by-product of the algorithm. As for MCMC, the performance of these algorithms is highly dependent on the target distributions $\{\pi_n\}_{n\in\mathbb{T}}$ and proposal distributions used to explore the space.

This paper focuses on the algorithmic aspects of SMC samplers. However, it is worth noting that our algorithms can be interpreted as interacting particle approximations of a Feynman-Kac flow in distribution space. Many general convergence results are available for these approximations and, consequently, for SMC samplers (Del Moral, 2004). Nevertheless, the SMC samplers developed here are such that many known estimates on the asymptotic behaviour of these general processes can be greatly improved. Several of these results can be found in Del Moral and Doucet (2003). In this article we provide the expressions for the asymptotic variances associated with central limit theorems.

The rest of the paper is organized as follows. In Section 2, we present a generic Sequential IS (SIS) algorithm to sample from a sequence of distributions $\{\pi_n\}_{n\in\mathbb{T}}$. We outline the limitations of this approach which severely restricts the way one can move the particles around the space. In Section 3, we provide a method to circumvent this problem by building an artificial sequence of joint distributions which admits fixed marginals. We provide guidelines for the design of efficient algorithms. Some extensions and connections with previous work are outlined. The remaining sections describe how to apply the SMC sampler methodology to two important special cases. Section 4 presents a generic approach to convert an MCMC sampler into an SMC sampler so as to sample from a fixed target distribution. This is illustrated on a Bayesian analysis of finite mixture distributions. Finally, Section 5 presents an application of SMC samplers to a sequential, trans-dimensional Bayesian inference problem. The proofs of the results in Section 3 can be found in the Appendix.

2. Sequential Importance Sampling

In this Section, we describe a generic iterative/sequential IS method to sample from a sequence of distributions $\{\pi_n\}_{n\in\mathbb{T}}$. We provide a review of the standard IS method, then we outline its limitations and describe a sequential version of the algorithm.

2.1. Importance Sampling

Let π_n be a target density on E such that

$$\pi_n(x) = \frac{\gamma_n(x)}{Z_n}$$

where $\gamma_n : E \to \mathbb{R}^+$ is known pointwise and the normalizing constant Z_n is unknown. Let $\eta_n(x)$ be a positive density with respect to $\mathrm{d}x$, referred to as the importance distribution. IS is based upon the following identities

$$\mathbb{E}_{\pi_n}(\varphi) = Z_n^{-1}\int_E \varphi(x) w_n(x)\,\eta_n(x)\,\mathrm{d}x, \tag{2}$$

$$Z_n = \int_E w_n(x)\eta_n(x)\mathrm{d}x, \tag{3}$$

where the unnormalized importance weight function is equal to

$$w_n(x) = \frac{\gamma_n(x)}{\eta_n(x)}. \tag{4}$$

By sampling N particles $\left\{X_n^{(i)}\right\}$ from η_n and substituting the Monte Carlo approximation

$$\eta_n^N(\mathrm{d}x) = \frac{1}{N}\sum_{i=1}^{N}\delta_{X_n^{(i)}}(\mathrm{d}x)$$

(with δ denoting Dirac measure) of this distribution into (2) and (3), we obtain an approximation of $\mathbb{E}_{\pi_n}(\varphi)$ and Z_n.

In statistics applications, we are typically interested in estimating (1) for a large range of test functions φ. In these cases, we are usually trying to select η_n 'close' to π_n as the variance is approximately proportional to $1+\text{var}_{\eta_n}[w_n(X_n)]$ (see Liu (2001) pp. 35-36). Unfortunately, selecting such an importance distribution is very difficult when π_n is a non standard high dimensional distribution. As a result, despite its relative simplicity, IS is almost never used when MCMC methods can be applied.

2.2. Sequential Importance Sampling

In order to obtain better importance distributions, we propose the following sequential method. At time $n = 1$, we start with a target distribution π_1 which is assumed easy to approximate efficiently using IS; that is, η_1 can be selected such that the variance of the importance weights (4) is small. In the simplest case, $\eta_1 = \pi_1$. Then at time $n = 2$, we consider the new target distribution π_2. To build the associated IS distribution η_2, we use the particles sampled at time $n = 1$, say $\left\{X_1^{(i)}\right\}$. The rationale is that if π_1 and π_2 are not too different from one another, then it should be possible to move the particles $\left\{X_1^{(i)}\right\}$ in the regions of high probability density of π_2 in a sensible way.

At time $n-1$ we have N particles $\left\{X_{n-1}^{(i)}\right\}$ distributed according to η_{n-1}. We propose to move these particles using a Markov kernel $K_n : E \times \mathcal{E} \to [0,1]$, with associated density denoted $K_n(x,x')$. The particles $\left\{X_n^{(i)}\right\}$ obtained this way are marginally distributed according to

$$\eta_n(x') = \int_E \eta_{n-1}(x) K_n(x,x') \, \mathrm{d}x. \tag{5}$$

If η_n *can be computed pointwise*, then it is possible to use the standard IS estimates of π_n and Z_n.

2.3. Algorithm Settings

This SIS strategy is very general. There are many potential choices for $\{\pi_n\}_{n\in\mathbb{T}}$ leading to various integration and optimization algorithms.

2.3.1. *Sequence of distributions* $\{\pi_n\}$.

- In the context of Bayesian inference for static parameters, where p observations $(y_1,\ldots,y_p)$ are available, one can consider

$$\pi_n(x) = p(x|\, y_1,\ldots,y_n). \tag{6}$$

See Chopin (2002) for such applications.

- It can be of interest to consider an inhomogeneous sequence of distributions to move 'smoothly' from a tractable distribution $\pi_1 = \mu_1$ to a target distribution π through a sequence of intermediate distributions. For example, one could select a geometric path (Gelman and Meng, 1998; Neal, 2001)

$$\pi_n(x) \propto [\pi(x)]^{\phi_n} [\mu_1(x)]^{1-\phi_n} \tag{7}$$

with $0 \leq \phi_1 < \cdots < \phi_p = 1$.

Alternatively, one could simply consider the case where $\pi_n = \pi$ for all $n \in \mathbb{T}$. This has been proposed numerous times in the literature. However, if π is a complex distribution, it is difficult to build a sensible initial importance distribution. In particular, such algorithms may fail when the target is multimodal with well-separated narrow modes. Indeed, in this case, the probability of obtaining samples in all the modes of the target is very small and an importance distribution based upon these initial particles is likely to be inefficient. Therefore, for difficult scenarios, it is unlikely that such approaches will be robust.

• For global optimization, as in simulated annealing, one can select

$$\pi_n(x) \propto [\pi(x)]^{\phi_n} \tag{8}$$

where $\{\phi_n\}_{n\in\mathbb{T}}$ an increasing sequence such that $\phi_p \rightarrow \infty$ for large p.

• Assume we are interested in estimating the probability of a rare event, $A \in \mathcal{E}$, under a probability measure π ($\pi(A) \approx 0$). In most of these applications, π is typically easy to sample from and the normalizing constant of its density is known. We can consider the sequence of distributions

$$\pi_n(x) \propto \pi(x)\, \mathbb{I}_{E_n}(x)$$

where $E_n \in \mathcal{E}\ \forall n \in \mathbb{T}$, $\mathbb{I}_A(x)$ is the indicator function for $A \in \mathcal{E}$ and $E_1 \supset E_2 \supset \cdots \supset E_{p-1} \supset E_p$, $E_1 = E$ and $E_p = A$. An estimate of $\pi(A)$ is given by an estimate of the normalizing constant Z_p.

2.3.2. *Sequence of transition kernels $\{K_n\}$.*

It is easily seen that the optimal proposal, in the sense of minimizing the variance of the importance weights, is $K_n(x, x') = \pi_n(x')$. As this choice is impossible, we must formulate sensible alternatives.

• *Independent proposals.* It is possible to select $K_n(x, x') = K_n(x')$ where $K_n(\cdot)$ is a standard distribution (e.g. Gaussian, multinomial) whose parameters can be determined using some statistics based upon η_{n-1}^N. This approach is standard in the literature; e.g. West (1993). However, independent proposals appear overly restrictive and it seems sensible to design local moves in high-dimensional cases.

• *Local random walk moves.* A standard alternative consists of using for $K_n(x, x')$ a random walk kernel. This idea has appeared several times in the literature where $K_n(x, x')$ is selected as a standard smoothing kernel (e.g. Gaussian, Epanechikov); e.g. Givens and Raftery (1996). However, this approach is problematic. Firstly, the choice of the kernel bandwidth is difficult. Standard rules to determine kernel bandwidths may indeed not be appropriate here, because we are not trying to obtain a kernel density estimate $\eta_{n-1}^N K_n(x')$ of $\eta_{n-1}(x')$ but to design an importance distribution to approximate $\pi_n(x')$. Secondly, no information about π_n is typically used to build $K_n(x, x')$.

Two alternative classes of local moves exploiting the structure of π_n are now proposed.

• *MCMC moves.* It is natural to set K_n as an MCMC kernel of invariant distribution π_n. In particular, this approach is justified if either K_n is fast mixing and/or π_n is slowly evolving so that one can expect η_n to be reasonably close to the target distribution. In this case, the resulting algorithm is an IS technique which would allow us to correct for the fact that the N inhomogeneous Markov chains $\left\{X_n^{(i)}\right\}$ are such that $\eta_n \neq \pi_n$. This is an attractive strategy: We are able to use the vast literature on the design of efficient MCMC algorithms to build 'good' importance distributions.

• *Approximate Gibbs moves.* When it is impossible to sample from the full conditional distributions required by a Gibbs kernel of invariant distribution π_n, an approximation of these distributions can be used to build K_n. This strategy is very popular in the SMC literature for optimal filtering where the so-called optimal proposal (Doucet et al., 2000, p. 199; Liu, 2001, p. 47) corresponds to a Gibbs step but can rarely be implemented and is approximated.

2.4. Limitations of Sequential Importance Sampling

For any probability density ν, we use the following notation

$$\nu K_{i:j}\left(x_j\right) \triangleq \int \nu\left(x_{i-1}\right) \prod_{k=i}^{j} K_k\left(x_{k-1}, x_k\right) \mathrm{d}x_{i-1:j-1}$$

where $x_{i:j}$, $i \leq j$, (resp. $X_{i:j}$) denotes $(x_i, \ldots, x_j)$ (resp. $(X_i, \ldots, X_j)$).

The algorithm discussed above suffers from a major drawback. In most cases, it is impossible to compute the importance distribution $\eta_n\left(x_n\right)$ given by

$$\eta_n\left(x_n\right) = \eta_1 K_{2:n}\left(x_n\right) \tag{9}$$

and hence impossible to compute the importance weights. An important exception is when one uses independent proposal distributions and, in our opinion, this explains why this approach is often used in the literature. However, whenever local moves are used, η_n does not admit a closed-form expression in most cases.

A potential solution is to attempt to approximate η_n pointwise by

$$\eta_{n-1}^N K_n\left(x_n\right) = \frac{1}{N} \sum_{i=1}^{N} K_n\left(X_{n-1}^{(i)}, x_n\right).$$

This approximation has been used in the literature for local random walk moves. However, this approach suffers from two major problems. First, the computational complexity of the resulting algorithm would be in $O\left(N^2\right)$ which is prohibitive. Second, it is impossible to compute $K_n\left(x_{n-1}, x_n\right)$ pointwise in important scenarios. For example, consider the case where $E = \mathbb{R}$, K_n is an MH kernel and $\mathrm{d}x$ is Lebesgue measure: We cannot, typically, compute the rejection probability of the MH kernel analytically.

3. SMC Samplers

In this Section, we show how it is possible to use any local move -including MCMC moves- in the SIS framework while circumventing the calculation of (9). The algorithm preserves complexity of $O\left(N\right)$ and provides asymptotically consistent estimates.

3.1. Methodology and Algorithm

As noted above, the importance weight can be computed exactly at time 1. At time $n > 1$, it is typically impossible to compute $\eta_n\left(x_n\right)$ pointwise as it requires an integration with respect to $x_{1:n-1}$. Instead, we propose an auxiliary variable technique and introduce artificial backward (in time) Markov kernels $L_{n-1} : E \times \mathcal{E} \rightarrow [0, 1]$ with density $L_{n-1}(x_n, x_{n-1})$. We then perform IS between the joint importance distribution $\eta_n(x_{1:n})$ and the artificial joint target distribution defined by

$$\widetilde{\pi}_n(x_{1:n}) = \frac{\widetilde{\gamma}_n(x_{1:n})}{Z_n}$$

where

$$\widetilde{\gamma}_n(x_{1:n}) = \gamma_n(x_n) \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k).$$

As $\widetilde{\pi}_n(x_{1:n})$ admits $\pi_n(x_n)$ as a marginal by construction, IS provides an estimate of this distribution and its normalizing constant. By proceeding thus, we have defined a sequence of probability distributions $\{\widetilde{\pi}_n\}$ whose dimension is increasing over time; i.e. $\widetilde{\pi}_n$ is defined on E^n. We are then back to the 'standard' SMC framework described, for example, in (Del Moral, 2004; Doucet *et al.*, 2001; Liu, 2001). We now describe a generic SMC algorithm to sample from this sequence of distributions based upon sequential IS resampling methodology.

At time $n-1$, assume a set of weighted particles $\left\{W_{n-1}^{(i)}, X_{1:n-1}^{(i)}\right\}$ $(i = 1, \ldots, N)$ approximating $\widetilde{\pi}_{n-1}$ is available,

$$\widetilde{\pi}_{n-1}^N(\mathrm{d}x_{1:n-1}) = \sum_{i=1}^N W_{n-1}^{(i)} \delta_{X_{1:n-1}^{(i)}}(\mathrm{d}x_{1:n-1}) \tag{10}$$

$$W_{n-1}^{(i)} = \frac{w_{n-1}(X_{1:n-1}^{(i)})}{\sum_{j=1}^N w_{n-1}(X_{1:n-1}^{(j)})}$$

At time n, we extend the path of each particle with a Markov kernel $K_n(x_{n-1}, x_n)$. Importance sampling is then used to correct for the discrepancy between the sampling distribution $\eta_n(x_{1:n})$ and $\widetilde{\pi}_n(x_{1:n})$. In this case the new expression for the unnormalized importance weights is given by

$$\begin{aligned} w_n(x_{1:n}) &= \frac{\widetilde{\gamma}_n(x_{1:n})}{\eta_n(x_{1:n})} \\ &= w_{n-1}(x_{1:n-1})\,\widetilde{w}_n(x_{n-1}, x_n) \end{aligned} \tag{11}$$

where the so-called (unnormalized) incremental weight $\widetilde{w}_n(x_{n-1}, x_n)$ is equal to

$$\widetilde{w}_n(x_{n-1}, x_n) = \frac{\gamma_n(x_n)\, L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1})\, K_n(x_{n-1}, x_n)}. \tag{12}$$

As the discrepancy between η_n and $\widetilde{\pi}_n$ tends to increase with n, the variance of the unnormalized importance weights tends to increase resulting in a potential degeneracy of the particle approximation. This degeneracy is routinely measured using the effective sample size (ESS) criterion $\left(\sum_{i=1}^N \left(W_n^{(i)}\right)^2\right)^{-1}$ (Liu and Chen, 1998). The ESS takes values between 1 and N. If the degeneracy is too high, i.e. the ESS is below a pre-specified threshold, say $N/2$, then each particle $X_{1:n}^{(i)}$ is copied $N_n^{(i)}$ times under the constraint $\sum_{i=1}^N N_n^{(i)} = N$; the expectation of $N_n^{(i)}$ being equal to $NW_n^{(i)}$ such that particles with high weights are copied multiple times whereas particles with low weights are discarded. Finally, all resampled particles are assigned equal weights. The simplest way to perform resampling consists of sampling the N new particles from the weighted distribution $\widetilde{\pi}_n^N$; the resulting $\left\{N_n^{(i)}\right\}$ are distributed according to a multinomial distribution of parameters $\left\{W_n^{(i)}\right\}$. Stratified resampling (Kitagawa, 1996) and residual resampling can also be used and all of these reduce the variance of $N_n^{(i)}$ relative to that of the multinomial scheme.

A summary of the algorithm can be found in Algorithm 1. The complexity of this algorithm is in $O\left(N\right)$ and it can be parallelized easily.

Algorithm 1 Sequential Monte Carlo Sampler.

```
1. INITIALIZATION
    • Set $n = 1$.
    • For $i = 1, \ldots, N$ draw $X_1^{(i)} \sim \eta_1$.
    • Evaluate $\{w_1(X_1^{(i)})\}$ using (4) and normalize these weights to obtain $\{W_1^{(i)}\}$.
Iterate steps 2. and 3.
2. RESAMPLING
    • If $\mathrm{ESS} < T$ (for some threshold $T$), resample the particles and set $W_n^{(i)} = 1/N$.
3. SAMPLING
    • Set $n = n + 1$, if $n = p + 1$ stop.
    • For $i = 1, \ldots, N$ draw $X_n^{(i)} \sim K_n(X_{n-1}^{(i)}, \cdot)$.
    • Evaluate $\{\widetilde{w}_n(X_{n-1:n}^{(i)})\}$ using (12) and normalize the weights
          $W_n^{(i)} = \frac{W_{n-1}^{(i)} \widetilde{w}_n(X_{n-1:n}^{(i)})}{\sum_{j=1}^{N} W_{n-1}^{(j)} \widetilde{w}_n(X_{n-1:n}^{(j)})}$
```

Remark. If the weights $\left\{W_n^{(i)}\right\}$ are independent of $\left\{X_n^{(i)}\right\}$, then the particles $\left\{X_n^{(i)}\right\}$ should be sampled after the weights $\left\{W_n^{(i)}\right\}$ have been computed and after the particle approximation $\left\{W_n^{(i)}, X_{n-1}^{(i)}\right\}$ of $\pi_n\left(x_{n-1}\right)$ has possibly been resampled. This scenario appears when $\{L_n\}$ is given by (30).

Remark. It is also possible to derive an auxiliary version of this algorithm in the spirit of (Pitt and Shephard, 1999).

3.2. *Notes on the Algorithm*

3.2.1. Estimates of Target Distributions and Normalizing Constants

At time n, we obtain after the sampling step a particle approximation $\left\{W_n^{(i)}, X_{1:n}^{(i)}\right\}$ of $\widetilde{\pi}_n\left(x_{1:n}\right)$. As the target $\pi_n\left(x_n\right)$ is a marginal of $\widetilde{\pi}_n\left(x_{1:n}\right)$ by construction, an approximation of it is given by

$$\pi_n^N\left(\mathrm{d}x\right)=\sum_{i=1}^{N}W_n^{(i)}\delta_{X_n^{(i)}}\left(\mathrm{d}x\right).$$

The particle approximation $\left\{W_{n-1}^{(i)},X_{n-1:n}^{(i)}\right\}$ of $\pi_{n-1}\left(x_{n-1}\right)K_n\left(x_{n-1},x_n\right)$ obtained after the sampling step also allows us to approximate

$$\frac{Z_n}{Z_{n-1}}=\frac{\int\gamma_n\left(x_n\right)\mathrm{d}x_n}{\int\gamma_{n-1}\left(x_{n-1}\right)\mathrm{d}x_{n-1}}\text{ by }\frac{\widehat{Z_n}}{Z_{n-1}}=\sum_{i=1}^{N}W_{n-1}^{(i)}\widetilde{w}_n\left(X_{n-1:n}^{(i)}\right). \tag{13}$$

To estimate Z_n/Z_1, one can use the product of estimates of the form (13) from time $k=2$ to n. However, if one does not resample at each iteration, a simpler alternative is given by

$$\frac{\widehat{Z_n}}{Z_1}=\prod_{j=1}^{r_{n-1}+1}\frac{\widehat{Z_{k_j}}}{Z_{k_{j-1}}},$$

with

$$\frac{\widehat{Z_{k_j}}}{Z_{k_{j-1}}}=\sum_{i=1}^{N}W_{k_{j-1}}^{(i)}\prod_{m=k_{j-1}+1}^{k_j}\widetilde{w}_m\left(X_{m-1:m}^{(i)}\right) \tag{14}$$

where $k_0=1$, k_j is the j^{th} time index at which one resamples for $j>1$. The number of resampling steps between 1 and $n-1$ is denoted r_{n-1} and we set $k_{r_{n-1}+1}=n$.

There is a potential alternative estimate for ratios of normalizing constants based upon path sampling (Gelman and Meng, 1998). Indeed, consider a continuous path of distributions

$$\pi_{\theta(t)}=\frac{\gamma_{\theta(t)}}{Z_{\theta(t)}}$$

where $t\in[0,1]$, $\theta\left(0\right)=0$ and $\theta\left(1\right)=1$. Then under regularity assumptions, we have the following path sampling identity

$$\log\frac{Z_1}{Z_0}=\int_0^1\frac{d\theta\left(t\right)}{dt}\int\frac{d\log\left(\gamma_{\theta(t)}\left(x\right)\right)}{dt}\pi_{\theta(t)}\left(\mathrm{d}x\right)dt. \tag{15}$$

In the SMC samplers context, if we consider a sequence of $p+1$ intermediate distributions denoted here $\pi_{\theta\left(\frac{k}{P}\right)}$ $k=0,\ldots,p$ to move from π_0 to π_1 then (15) can be approximated using a trapezoidal integration scheme and substituting $\widehat{\pi}_{\theta\left(\frac{k}{P}\right)}^N\left(\mathrm{d}x\right)$ to $\pi_{\theta\left(\frac{k}{P}\right)}\left(\mathrm{d}x\right)$. Some applications of this identity in an SMC framework are detailed in Johansen *et al.* (2005) and Rousset and Stoltz (2005).

3.2.2. *Mixture of Markov Kernels*

The algorithm described in this section must be interpreted as the basic element of more complex algorithms. It is to SMC what the MH algorithm is to MCMC. For complex MCMC problems, one typically uses a combination of MH steps where the J components of x say $(x_1,\ldots,x_J)$ are updated in sub-blocks. Similarly, to sample from high dimensional distributions, a practical SMC sampler can update the components of x via sub-blocks and a mixture of transition kernels can be used at each time n.

Let us assume $K_n(x_{n-1},x_n)$ is of the form

$$K_n(x_{n-1},x_n)=\sum_{m=1}^{M}\alpha_{n,m}(x_{n-1})\,K_{n,m}(x_{n-1},x_n) \tag{16}$$

where $\alpha_{n,m}(x_{n-1})\geq 0$, $\sum_{m=1}^{M}\alpha_{n,m}(x_{n-1})=1$ and $\{K_{n,m}\}$ is a collection of transition kernels. In this case, the incremental weights can be computed by the standard formula (12). However, this could be too expensive if M is large. An alternative, valid, approach consists of considering a backward Markov kernel of the form

$$L_{n-1}(x_n,x_{n-1})=\sum_{m=1}^{M}\beta_{n-1,m}(x_n)\,L_{n-1,m}(x_n,x_{n-1}) \tag{17}$$

where $\beta_{n-1,m}(x_n)\geq 0$, $\sum_{m=1}^{M}\beta_{n-1,m}(x_n)=1$ and $\{L_{n-1,m}\}$ is a collection of backward transition kernels. We now introduce, explicitly, a discrete latent variable M_n taking values in $\mathcal{M}=\{1,\ldots,M\}$ such that $\mathbb{P}(M_n=m)=\alpha_{n,m}(x_{n-1})$ and perform IS on the extended space $E\times E\times\mathcal{M}$. This yields an incremental importance weight equal to

$$\widetilde{w}_n(x_{n-1},x_n,m_n)=\frac{\gamma_n(x_n)\,\beta_{n-1,m_n}(x_n)\,L_{n-1,m_n}(x_n,x_{n-1})}{\gamma_{n-1}(x_{n-1})\,\alpha_{n,m_n}(x_{n-1})\,K_{n,m_n}(x_{n-1},x_n)}. \tag{18}$$

The variance of (18) will always be superior or equal to the variance of (12).

3.3. *Algorithm Settings*

3.3.1. *Optimal Backward Kernels*

In standard applications of SMC methods, only the proposal kernels $\{K_n\}$ have to be selected as the joint distributions $\{\widetilde{\pi}_n\}$ are given by the problem at hand. In the framework considered here, $\{L_n\}$ is arbitrary. However, in practice, $\{L_n\}$ should be optimized with respect to $\{K_n\}$ in order to obtain good performance. Recall that $\{L_n\}$ has been introduced because it was impossible to compute the marginal importance distribution $\{\eta_n\}$ pointwise.

The marginal distribution of the particles $\left\{X_n^{(i)}\right\}$ at time n is given by

$$\eta_n(x_n)=\eta_1 K_{2:n}(x_n) \tag{19}$$

if the particles have not been resampled before time n and approximately

$$\eta_n(x_n)=\pi_l K_{l+1:n}(x_n) \tag{20}$$

if the last time the particles were resampled was l. To simplify the discussion, we consider here the case (19), note that the more general case (20) can be handled similarly.

The introduction of the auxiliary kernels $\{L_n\}$ means that we need not compute $\eta_n(x_n)$. This comes at the price of extending the integration domain from E to E^n and increasing the variance (if it exists) of the importance weights. The following proposition establishes the expression of the sequence of optimal backward Markov kernels.

Proposition 3.1. *The sequence of kernels $\left\{L_k^{opt}\right\}$ $(k=1,\ldots,n)$ minimizing the variance of the unnormalized importance weight $w_n(x_{1:n})$ is given for any k,n by*

$$L_{k-1}^{opt}(x_k,x_{k-1})=\frac{\eta_{k-1}(x_{k-1})\,K_k(x_{k-1},x_k)}{\eta_k(x_k)} \tag{21}$$

and in this case

$$w_n\left(x_{1:n}\right)=\frac{\gamma_n\left(x_n\right)}{\eta_n\left(x_n\right)}.$$

Remark. This proposition is intuitive and simply states that the optimal backward Markov kernels take us back to the case where one performs importance sampling on E instead of E^n. Note that the result can also be intuitively understood through the following forward-backward formula for Markov processes

$$\eta_1\left(x_1\right)\prod_{k=2}^{n}K_k\left(x_{k-1},x_k\right)=\eta_n\left(x_n\right)\prod_{k=2}^{n}L_{k-1}^{\text{opt}}\left(x_k,x_{k-1}\right). \tag{22}$$

In the context of a mixture of kernels (16), one can use Proposition 3.1 to establish that the optimal backward kernel is of the form (17) with

$$\beta_{n-1,m}^{\text{opt}}\left(x_n\right)\propto\int\alpha_{n,m}\left(x_{n-1}\right)\eta_{n-1}\left(x_{n-1}\right)K_n\left(x_{n-1},x_n\right)\mathrm{d}x_{n-1}, \tag{23}$$

$$L_{n-1,m}^{\text{opt}}\left(x_n,x_{n-1}\right)=\frac{\alpha_{n,m}\left(x_{n-1}\right)\eta_{n-1}\left(x_{n-1}\right)K_n\left(x_{n-1},x_n\right)}{\int\alpha_{n,m}\left(x_{n-1}\right)\eta_{n-1}\left(x_{n-1}\right)K_n\left(x_{n-1},x_n\right)\mathrm{d}x_{n-1}}. \tag{24}$$

3.3.2. Sub-Optimal Backwards Kernels

It is typically impossible, in practice, to use the optimal kernel as they themselves rely on marginal distributions which do not admit any closed-form expression. However, this suggests that we should select $\{L_k\}$ to approximate (21). The key point is that, even if $\{L_k\}$ is different from (21), the algorithm will still provide asymptotically consistent estimates. Some approximations are now discussed.

• *Substituting* π_{n-1} *for* η_{n-1}. One point used recurrently is that (12) suggests that a sensible, sub-optimal, strategy consists of using an L_n which is an approximation of the optimal kernel (21) where one has substituted π_{n-1} for η_{n-1}, that is:

$$L_{n-1}\left(x_n,x_{n-1}\right)=\frac{\pi_{n-1}\left(x_{n-1}\right)K_n\left(x_{n-1},x_n\right)}{\pi_{n-1}K_n\left(x_n\right)} \tag{25}$$

which yields

$$\widetilde{w}_n\left(x_{n-1},x_n\right)=\frac{\gamma_n\left(x_n\right)}{\int_E\gamma_{n-1}\left(x_{n-1}\right)K_n\left(x_{n-1},x_n\right)\mathrm{d}x_{n-1}}. \tag{26}$$

It is often more convenient to use (26) than (21) as $\{\gamma_n\}$ is known analytically, whilst $\{\eta_n\}$ is not. It should be noted that, if particles have been resampled at time $n-1$, then η_{n-1} is indeed approximately equal to π_{n-1} and thus (21) is equal to (25).

• *Gibbs Type Updates.* Consider the case where $x=(x_1,\ldots,x_J)$ and we only want to update the k^{th} $(k\in\{1,...,J\})$ component x_k of x denoted $x_{n,k}$ at time n. It is straightforward to establish that the proposal distribution minimizing the variance of (26) conditional upon x_{n-1} is a Gibbs update; i.e.

$$K_n\left(x_{n-1},\mathrm{d}x_n\right)=\delta_{x_{n-1,-k}}\left(\mathrm{d}x_{n,-k}\right)\pi_n\left(\mathrm{d}x_{n,k}\middle|\,x_{n,-k}\right) \tag{27}$$

where $x_{n,-k}=(x_{n,1},\ldots,x_{n,k-1},x_{n,k+1},\ldots,x_{n,J})$. In this case (25) and (26) are given by

$$L_{n-1}\left(x_n,\mathrm{d}x_{n-1}\right)=\delta_{x_{n,-k}}\left(\mathrm{d}x_{n-1,-k}\right)\pi_{n-1}\left(\mathrm{d}x_{n-1,k}\middle|\,x_{n-1,-k}\right),$$

$$\widetilde{w}_n\left(x_{n-1}, x_n\right)=\frac{\gamma_n\left(x_{n-1,-k}, x_{n,k}\right)}{\gamma_{n-1}\left(x_{n-1,-k}\right)\pi_n\left(\left.x_{n,k}\right| x_{n-1,-k}\right)}.$$

When it is not possible to sample from $\pi_n\left(\left.x_{n,k}\right| x_{n-1,-k}\right)$ and/or compute $\gamma_{n-1}\left(x_{n-1,-k}\right)=\int\gamma_{n-1}\left(x_{n-1}\right)\mathrm{d}x_{n-1,k}$ analytically, this suggests using an approximation $\widehat{\pi}_n\left(\left.x_{n,k}\right| x_{n-1,-k}\right)$ of $\pi_n\left(\left.x_{n,k}\right| x_{n-1,-k}\right)$ to sample the particles and another approximation $\widehat{\pi}_{n-1}\left(\left.x_{n-1,k}\right| x_{n-1,-k}\right)$ of $\pi_{n-1}\left(\left.x_{n-1,k}\right| x_{n-1,-k}\right)$ to obtain

$$L_{n-1}\left(x_n, \mathrm{d}x_{n-1}\right)=\delta_{x_{n,-k}}\left(\mathrm{d}x_{n-1,-k}\right)\widehat{\pi}_{n-1}\left(\left.\mathrm{d}x_{n-1,k}\right| x_{n-1,-k}\right), \tag{28}$$

$$\widetilde{w}_n\left(x_{n-1}, x_n\right)=\frac{\gamma_n\left(x_{n-1,-k}, x_{n,k}\right)\widehat{\pi}_{n-1}\left(\left.x_{n-1,k}\right| x_{n-1,-k}\right)}{\gamma_{n-1}\left(x_{n-1}\right)\widehat{\pi}_n\left(\left.x_{n,k}\right| x_{n-1,-k}\right)}. \tag{29}$$

• *MCMC Kernels.* A generic alternative approximation of (25) can also be made when K_n is an MCMC kernel of invariant distribution π_n. It is given by

$$L_{n-1}\left(x_n, x_{n-1}\right)=\frac{\pi_n\left(x_{n-1}\right)K_n\left(x_{n-1}, x_n\right)}{\pi_n\left(x_n\right)} \tag{30}$$

and will be a good approximation of (25) if $\pi_{n-1}\approx\pi_n$; note that (30) is the reversal Markov kernel associated with K_n. In this case, one has unnormalized incremental weight

$$\widetilde{w}_n\left(x_{n-1}, x_n\right)=\frac{\gamma_n\left(x_{n-1}\right)}{\gamma_{n-1}\left(x_{n-1}\right)}. \tag{31}$$

Contrary to (25), this approach does not apply in scenarios where $E_{n-1}\subset E_n$ and $E_n\in\mathcal{E}\ \forall n\in\mathbb{T}$ as discussed in Section 5. Indeed, in this case

$$L_{n-1}\left(x_n, x_{n-1}\right)=\frac{\pi_n\left(x_{n-1}\right)K_n\left(x_{n-1}, x_n\right)}{\int_{E_{n-1}}\pi_n\left(x_{n-1}\right)K_n\left(x_{n-1}, x_n\right)\mathrm{d}x_{n-1}} \tag{32}$$

but the denominator of this expression is different from $\pi_n\left(x_n\right)$ as the integration is over E_{n-1} and not E_n.

• *Mixtures of Kernels.* Practically, one cannot typically compute the expressions (23) and (24) in closed form and so approximations are also necessary. As discussed previously, one sub-optimal choice consists of replacing η_{n-1} with π_{n-1} in (23) and (24) or use further approximations like (30).

3.3.3. *Summary*

To conclude this subsection, we emphasize that selecting $\{L_n\}$ as close as possible to $\{L_n^{\text{opt}}\}$ is crucial for this method to be efficient. It could be tempting to select $\{L_n\}$ in a different way. For example, if we select $L_{n-1}=K_n$ then the incremental importance weight looks like a MH ratio. However, this 'aesthetic' choice will be inefficient in most cases resulting in importance weights with a very large or infinite variance.

3.4. *Convergence Results*

Using (10), the SMC algorithm yields estimates of expectations (1) via

$$\mathbb{E}_{\pi_n^N}\left(\varphi\right)=\int_E\varphi\left(x\right)\pi_n^N\left(\mathrm{d}x\right). \tag{33}$$

Using (13), we can also obtain an estimate of $\log\left(Z_n/Z_1\right)$

$$\log \widehat{\frac{Z_n}{Z_1}} = \sum_{k=2}^{n} \log \widehat{\frac{Z_k}{Z_{k-1}}}. \tag{34}$$

We now present a central limit theorem, giving the asymptotic variance of these estimates in two 'extreme' cases: when we never resample and when we resample at each iteration. For the sake of simplicity, we have only considered the case where multinomial resampling is used (see Chopin (2004a) for analysis using residual resampling and also Künsch (2005) for results in the context of filtering). The asymptotic variance expressions of (33) and (34) for general SMC algorithms have previously been established in the literature. However, we propose here a new representation which clarifies the influence of the kernels $\{L_n\}$.

In the following proposition, we denote by $\mathcal{N}(\mu,\sigma^2)$ the Normal distribution with mean μ and variance σ^2, convergence in distribution by '$\Rightarrow$', $\int \widetilde{\pi}_n\left(x_{1:n}\right) \mathrm{d}x_{1:k-1}\, \mathrm{d}x_{k+1:n}$ by $\widetilde{\pi}_n\left(x_k\right)$ and $\int \widetilde{\pi}_n\left(x_{1:n}\right) \mathrm{d}x_{1:k-1}\, \mathrm{d}x_{k+1:n-1}/\ \widetilde{\pi}_n\left(x_k\right)$ by $\widetilde{\pi}_n\left(\left.x_n\right| x_k\right)$.

Proposition 3.2. *Under the weak integrability conditions given in (Chopin, 2004; Theorem 1) or (Del Moral, 2004, Section 9.4, pp. 300-306), one obtains the following results. When no resampling is performed, one has*

$$\sqrt{N}\left(\mathbb{E}_{\pi_n^N}\left(\varphi\right) - \mathbb{E}_{\pi_n}\left(\varphi\right)\right) \Rightarrow \mathcal{N}\left(0, \sigma^2_{IS,n}\left(\varphi\right)\right)$$

with

$$\sigma^2_{IS,n}\left(\varphi\right) = \int \frac{\widetilde{\pi}_n\left(x_{1:n}\right)^2}{\eta_n\left(x_{1:n}\right)} \left(\varphi\left(x_n\right) - \mathbb{E}_{\pi_n}\left(\varphi\right)\right)^2 \mathrm{d}x_{1:n} \tag{35}$$

where the joint importance distribution η_n is given by

$$\eta_n\left(x_{1:n}\right) = \eta_1\left(x_1\right) \prod_{k=2}^{n} K_k\left(x_{k-1}, x_k\right).$$

We also have

$$\sqrt{N}\left(\log \widehat{\frac{Z_n}{Z_1}} - \log \frac{Z_n}{Z_1}\right) \Rightarrow \mathcal{N}\left(0, \sigma^2_{IS,n}\right)$$

with

$$\sigma^2_{IS,n} = \int \frac{\widetilde{\pi}_n\left(x_{1:n}\right)^2}{\eta_n\left(x_{1:n}\right)} \mathrm{d}x_{1:n} - 1. \tag{36}$$

When multinomial resampling is used at each iteration, one has

$$\sqrt{N}\left(\mathbb{E}_{\pi_n^N}\left(\varphi\right) - \mathbb{E}_{\pi_n}\left(\varphi\right)\right) \Rightarrow \mathcal{N}\left(0, \sigma^2_{SMC,n}\left(\varphi\right)\right)$$

where, for $n \geq 2$,

$$\begin{aligned}
&\sigma^2_{SMC,n}\left(\varphi\right) \\
&= \int \frac{\widetilde{\pi}_n\left(x_1\right)^2}{\eta_1\left(x_1\right)} \left(\int \varphi\left(x_n\right) \widetilde{\pi}_n\left(\left.x_n\right| x_1\right) \mathrm{d}x_n - \mathbb{E}_{\pi_n}\left(\varphi\right)\right)^2 \mathrm{d}x_1 \\
&+ \sum_{k=2}^{n-1} \int \frac{\left(\widetilde{\pi}_n\left(x_k\right) L_{k-1}\left(x_k, x_{k-1}\right)\right)^2}{\pi_{k-1}\left(x_{k-1}\right) K_k\left(x_{k-1}, x_k\right)} \left(\int \varphi\left(x_n\right) \widetilde{\pi}_n\left(\left.x_n\right| x_k\right) \mathrm{d}x_n - \mathbb{E}_{\pi_n}\left(\varphi\right)\right)^2 \mathrm{d}x_{k-1:k} \\
&+ \int \frac{\left(\pi_n\left(x_n\right) L_{n-1}\left(x_n, x_{n-1}\right)\right)^2}{\pi_{n-1}\left(x_{n-1}\right) K_n\left(x_{n-1}, x_n\right)} \left(\varphi\left(x_n\right) - \mathbb{E}_{\pi_n}\left(\varphi\right)\right)^2 \mathrm{d}x_{n-1:n}
\end{aligned} \tag{37}$$

and

$$\sqrt{N}\left(\log\frac{\widehat{Z_n}}{Z_1}-\log\frac{Z_n}{Z_1}\right)\Rightarrow\mathcal{N}\left(0,\sigma^2_{SMC,n}\right)$$

where

$$\begin{aligned}\sigma^2_{SMC,n} &= \int\frac{\widetilde{\pi}_n\left(x_1\right)^2}{\eta_1\left(x_1\right)}\mathrm{d}x_1-1 \\ &+\sum_{k=2}^{n-1}\left(\int\frac{\left(\widetilde{\pi}_n\left(x_k\right)L_{k-1}\left(x_k,x_{k-1}\right)\right)^2}{\pi_{k-1}\left(x_{k-1}\right)K_k\left(x_{k-1},x_k\right)}\mathrm{d}x_{k-1:k}-1\right) \\ &+\int\frac{\left(\pi_n\left(x_n\right)L_{n-1}\left(x_n,x_{n-1}\right)\right)^2}{\pi_{n-1}\left(x_{n-1}\right)K_n\left(x_{n-1},x_n\right)}\mathrm{d}x_{n-1:n}-1.\end{aligned} \tag{38}$$

Remark. In the general case, we cannot claim that $\sigma^2_{SMC,n}(\varphi)<\sigma^2_{IS,n}(\varphi)$ or $\sigma^2_{SMC,n}<\sigma^2_{IS,n}$. This is because, if the importance weights do not have a large variance, resampling is typically wasteful as any resampling scheme introduces some variance. However, resampling is beneficial in cases where successive distributions can vary significantly. This has been established theoretically in the filtering case in (Chopin, 2004; Theorem 5): Under mixing assumptions, the variance is shown to be upper bounded uniformly in time with resampling and to go to infinity without it. The proof may adapted to the class of problems considered here, and it can be shown that for (8) - under mixing assumptions on $\{K_n\}$ and using (25) or (30) for $\{L_n\}$ - the variance $\sigma^2_{SMC,n}(\varphi)$ is upper bounded uniformly in time for a logarithmic schedule $\{\phi_n\}$ whereas $\sigma^2_{IS,n}(\varphi)$ goes to infinity with n. Similar results hold for residual resampling. Finally we note that, although the resampling step appears somewhat artificial in discrete time, it appears naturally in the continuous time version of these algorithms (Del Moral, 2004; Rousset and Stoltz, 2005).

3.5. Connections to other work

To illustrate the connections with, and differences to, other work published in the literature, let us consider the case where we sample from $\{\pi_n\}$ using MCMC kernels $\{K_n\}$ where K_n is π_n-invariant.

Suppose, at time $n-1$, we have the particle approximation $\left\{W^{(i)}_{n-1},X^{(i)}_{n-1}\right\}$ of π_{n-1}. Several recent algorithms are based upon the implicit or explicit use of the backward kernel (30). In the case addressed here, where all the target distributions are defined on the same space, it is used for example in: Chopin (2002), Jarzynski (1997) and Neal (2001). In the case where the dimension of the target distributions increases over time, it is used in Gilks and Berzuini (2001) and MacEachern *et al.* (1999).

For the algorithms listed above, the associated backward kernels lead to the incremental weights:

$$\widetilde{w}_n\left(X^{(i)}_{n-1},X^{(i)}_n\right)\propto\frac{\pi_n\left(X^{(i)}_{n-1}\right)}{\pi_{n-1}\left(X^{(i)}_{n-1}\right)}. \tag{39}$$

The potential problem with (39) is that these weights are independent of $\left\{X^{(i)}_n\right\}$ where $X^{(i)}_n\sim K_n\left(X^{(i)}_{n-1},\cdot\right)$. In particular, the variance of (39) will typically be high if the discrepancy between π_{n-1} and π_n is large even if the kernel K_n mixes very well. This result is counter-intuitive. In the context of AIS (Neal, 2001) where the sequence of p target distributions (7) is supposed to satisfy $\pi_{n-1} \approx \pi_n$, this is not a problem. However, if successive distributions vary significantly, as in sequential Bayesian estimation, this can become a significant problem. For example, in the limiting case where $K_n(x_{n-1}, x_n) = \pi_n(x_n)$, one would end up with a particle approximation $\left\{W_n^{(i)}, X_n^{(i)}\right\}$ of π_n where the weights $\left\{W_n^{(i)}\right\}$ have an high variance whereas $\left\{X_n^{(i)}\right\}$ are i.i.d samples from π_n; this is clearly suboptimal.

To deal with the above problem, RM strategies are used by (among others) Chopin (2002) and Gilks and Berzuini (2001). RM corresponds to the SMC algorithm described in Section 3 using the backward kernel (30). RM resamples the particle approximation $\left\{W_n^{(i)}, X_{n-1}^{(i)}\right\}$ of $\widetilde{\pi}_n(x_{n-1})$ (if the variance of $\left\{W_n^{(i)}\right\}$ measured approximately through the ESS is high) and only then do we sample $\left\{X_n^{(i)}\right\}$ to obtain a particle approximation $\left\{N^{-1}, X_n^{(i)}\right\}$ of π_n; i.e. all particles have an equal weight. This can be expected to improve over not resampling if consecutive targets differ significantly and the kernels $\{K_n\}$ mix reasonably well; we demonstrate this in Section 4.

Proposition 3.1 suggests that a better choice (than (30)) of backward kernels is given by (25) for which the incremental weights are given by

$$\widetilde{w}_n\left(X_{n-1}^{(i)}, X_n^{(i)}\right) \propto \frac{\pi_n\left(X_n^{(i)}\right)}{\pi_{n-1}K_n\left(X_n^{(i)}\right)}. \tag{40}$$

The expression of (40) is much more intuitive than (39). It depends on K_n and thus the expression of the weights (40) reflects the mixing properties of the kernel K_n. In particular, the variance of (40) decreases as the mixing properties of the kernel increases.

To illustrate the difference between SMC using (40) instead of (39), consider the case where $x = (x_1, \ldots, x_J)$ and we use the Gibbs kernel (27) to update the component x_k so that (40) is given by

$$\widetilde{w}_n\left(X_{n-1}^{(i)}, X_n^{(i)}\right) \propto \frac{\pi_n\left(X_{n-1,-k}^{(i)}\right)}{\pi_{n-1}\left(X_{n-1,-k}^{(i)}\right)}. \tag{41}$$

By a simple Rao-Blackwell argument, the variance of (41) is always smaller than the variance of (39). The difference will be particularly significant in scenarios where the marginals $\pi_{n-1}(x_{-k})$ and $\pi_n(x_{-k})$ are close to each other but the full conditional distributions $\pi_n(x_k|\, x_{-k})$ and $\pi_{n-1}(x_k|\, x_{-k})$ differ significantly. In such cases, SMC using (39) resamples many more times than SMC using (41). Such scenarios appear for example in sequential Bayesian inference as described in Section 5 where each new observation only modifies the distribution of a subset of the variables significantly.

It is, unfortunately, not always possible to use (25) instead of (30) as an integral appears in (40). However, if the full conditional distributions of π_{n-1} and π_n can be approximated analytically, it is possible to use (28)-(29) instead.

Recent work of Cappé *et al.* (2004) is another special case of the proposed framework. The authors consider the homogeneous case where $\pi_n = \pi$ and $L_n(x, x') = \pi(x')$. Their algorithm corresponds to the case where $K_n(x, x') = K_n(x')$ and the parameters of $K_n(x')$

are determined using statistics over the entire population of particles at time $n-1$. Extensions of this work for missing data problems are presented in Celeux *et al.* (2006).

Finally Liang (2002) presents a related algorithm where $\pi_n = \pi$, $K_n(x, x') = L_n(x, x') = K(x, x')$.

4. From MCMC to SMC

4.1. Methodology

We now summarize how it is possible to obtain an SMC algorithm to sample from a fixed target distribution, π, using MCMC kernels or approximate Gibbs steps to move the particles around the space. The procedure is:

- Build a sequence of distributions $\{\pi_n\}$, $n = 1, \ldots, p$, such that π_1 is easy to sample from/approximate and $\pi_p = \pi$,
- Build a sequence of MCMC transition kernels $\{K_n\}$ such that K_n is π_n-invariant or K_n is an approximate Gibbs move of invariant distribution π_n,
- Based upon $\{\pi_n\}$ and $\{K_n\}$, build a sequence of artificial backward Markov kernels $\{L_n\}$ approximating $\{L_n^{\text{opt}}\}$. Two generic choices are (25) and (30). For approximate Gibbs moves, we can use (28).
- Use the SMC algorithm described in the previous section to approximate $\{\pi_n\}$ and estimate $\{Z_n\}$.

4.2. Bayesian Analysis of Finite Mixture Distributions

In the following example, we consider a mixture modelling problem. Our objective is to illustrate the potential benefits of resampling in the SMC methodology.

4.2.1. Model

Mixture models are typically used to model heterogeneous data, or as a simple means of density estimation; see Richardson and Green (1997) and the references therein for an overview. Bayesian analysis of mixtures has been fairly recent and there is often substantial difficulty in simulation from posterior distributions for such models; see Jasra *et al.* (2005b) for example.

We use the model of Richardson and Green (1997), which is as follows; data $y_1, \ldots, y_c$ are i.i.d with distribution

$$y_i|\theta_r \sim \sum_{j=1}^{r} \omega_j \mathcal{N}(\mu_j, \lambda_j^{-1})$$

where $\theta_r = (\mu_{1:r}, \lambda_{1:r}\ , \omega_{1:r})$, $2 \leq r < \infty$ and r known. The parameter space is $E = \mathbb{R}^r \times (\mathbb{R}^+)^r \times \mathcal{S}_r$ for the $r-$component mixture model where $\mathcal{S}_r = \{\omega_{1:r} : 0 \leq \omega_j \leq 1 \cap \sum_{j=1}^{r} \omega_j = 1\}$. The priors, which are the same for each component $j = 1, \ldots, r$, are taken to be: $\mu_j \sim \mathcal{N}(\xi, \kappa^{-1})$, $\lambda_j \sim \mathcal{G}a(\nu, \chi)$, $\omega_{1:r-1} \sim \mathcal{D}(\rho)$, where $\mathcal{D}(\rho)$ is the Dirichlet distribution with parameter ρ and $\mathcal{G}a(\nu, \chi)$ is the Gamma distribution with shape ν and scale χ. We set the priors in an identical manner to Richardson and Green (1997), with the χ parameter set as the mean of the hyper-prior they assigned that parameter.

One particular aspect of this model, which makes it an appropriate test example, is the feature of label switching. As noted above, the priors on each component are exchangeable, and consequently, in the posterior, the marginal distribution of μ_1 is the same as μ_2. That is, the marginal posterior is equivalent for each component specific quantity. This provides us with a diagnostic to establish the effectiveness of the simulation procedure. For more discussion see, for example, Jasra *et al.* (2005b). It should be noted that very long runs of an MCMC sampler targeting π_p were unable to explore all the modes of this distribution and failed to produce correct estimates (see Jasra *et al.* (2005b)).

4.2.2. SMC Sampler

We will consider AIS and SMC samplers. Both algorithms use the same MCMC kernels K_n with invariant distribution π_n and the same backward kernels (30). The MCMC kernel is a composition of the following update steps:

(a) Update $\mu_{1:r}$ via a MH kernel with additive normal random walk proposal.
(b) Update $\lambda_{1:r}$ via a MH kernel with multiplicative log-normal random walk proposal.
(c) Update $\omega_{1:r}$ via a MH kernel with additive normal random walk proposal on the logit scale.

For some of the runs of the algorithm, we will allow more than one iteration of the above Markov kernel per time step. Finally, the sequence of densities is taken as

$$\pi_n(\theta_r) \propto l(y_{1:c}; \theta_r)^{\phi_n} f(\theta_r)$$

where $0 \leq \phi_1 < \cdots < \phi_p = 1$ are tempering parameters and we have denoted the prior density as f and likelihood function as l.

4.2.3. Illustration

• *Data & Simulation Parameters.* For the comparison, we used the simulated data from Jasra *et al.* (2005b): 100 simulated data points from an equally weighted mixture of 4 (i.e. $r = 4$) normal densities with means at (-3,0,3,6) and standard deviations 0.55. We ran SMC samplers and AIS with MCMC kernels with invariant distribution π_n for 50, 100, 200, 500 and 1000 time steps with 1 and 10 MCMC iterations per time step. The proposal variances for the MH steps were the same for both procedures and were dynamically falling to produce an average acceptance rate in $(0.15, 0.6)$. The initial importance distribution was the prior. The C^{++} code and the data are available at the following address `http://www.cs.ubc.ca/~arnaud/smcsamplers.html`.

We ran the SMC algorithm with $N = 1000$ particles and we ran AIS for a similar CPU time. The absence of a resampling step allows AIS to run for a few more iterations than SMC. We ran each sampler 10 times (i.e. for each time specification and iteration number, each time with 1000 particles). For this demonstration, the resampling threshold was 500 particles. We use systematic resampling. The results with residual resampling are very similar.

We selected a piecewise linear cooling schedule $\{\phi_n\}$. Over 1000 time steps, the sequence increased uniformly from 0 to 15/100 for the first 200 time points then from 15/100 to 40/100 for the next 400 and finally from 40/100 to 1 for the last 400 time points. The other time specifications had the same proportion of time attributed to the tempering parameter setting. The choice was made to allow an initially slow evolution of the densities and then to allow more complex densities to appear at a faster rate. We note that other cooling

Table 1. Results from Mixture Comparison for SMC and AIS; We ran each sampler 10 times with 1000 particles. For AIS the number of time steps is slightly higher than stated, as it corresponds to the same CPU time as SMC.

Sampler Details	**Iterations per time step**	
SMC (50 time steps)	1	10
Avg. Log Posterior	-155.22	-152.03
Avg. Times Resampled	7.70	10.90
Avg. Log Normalizing Constant	-245.86	-240.90
AIS (50 time steps)		
Avg. Log Posterior	-191.07	-166.73
Avg. Log Normalizing Constant	-249.04	-242.07
SMC (100 time steps)		
Avg. Log Posterior	-153.08	-152.97
Avg. Times Resampled	8.20	5.10
Avg. Log Normalizing Constant	-245.43	-244.18
AIS (100 time steps)		
Avg. Log Posterior	-180.76	-162.37
Avg. Log Normalizing Constant	-250.22	-244.17
SMC (200 time steps)		
Avg. Log Posterior	-152.62	-152.99
Avg. Times Resampled	8.30	4.20
Avg. Log Normalizing Constant	-246.22	-245.84
AIS (200 time steps)		
Avg. Log Posterior	-174.40	-160.00
Avg. Log Normalizing Constant	-247.45	-245.92
SMC (500 time steps)	1	10
Avg. Log Posterior	-152.31	-151.90
Avg. Times Resampled	7.00	3.00
Avg. Log Normalizing Constant	-247.08	-247.01
AIS (500 time steps)		
Avg. Log Posterior	-167.67	-157.06
Avg. Log Normalizing Constant	-247.30	-247.94
SMC (1000 time steps)		
Avg. Log Posterior	-152.12	-151.94
Avg. Times Resampled	5.70	2.00
Avg. Log Normalizing Constant	-247.40	-247.40
AIS (1000 time steps)		
Avg. Log Posterior	-163.14	-155.31
Avg. Log Normalizing Constant	-247.50	-247.36

Table 2. Estimates of Means from Mixture Comparison for SMC and AIS. We ran each sampler 10 times with 1000 particles. The estimates are presented in increasing order, for presentation purposes.

Sampler Details	**Component**			
	1	2	3	4
SMC (50 steps, 1 iteration)	0.38	0.83	1.76	2.69
AIS (50 steps, 1 iteration)	0.03	0.75	1.68	2.28
SMC (50 steps, 10 iterations)	1.06	1.39	1.62	1.70
AIS (50 steps, 10 iterations)	0.26	0.96	1.61	2.85
SMC (100 steps, 1 iteration)	0.68	0.91	2.02	2.14
AIS (100 steps, 1 iteration)	0.61	0.75	1.46	2.72
SMC (100 steps, 10 iterations)	1.34	1.44	1.44	1.54
AIS (100 steps, 10 iterations)	0.88	1.06	1.59	2.25
SMC (200 steps, 1 iteration)	1.11	1.29	1.39	1.98
AIS (200 steps, 1 iteration)	0.89	1.23	1.72	1.96
SMC (200 steps, 10 iterations)	1.34	1.37	1.53	1.53
AIS (200 steps, 10 iterations)	1.26	1.34	1.45	1.74
SMC (500 steps, 1 iteration)	0.98	1.38	1.54	1.87
AIS (500 steps, 1 iteration)	0.87	1.31	1.47	2.12
SMC (500 steps, 10 iterations)	1.40	1.44	1.42	1.50
AIS (500 steps, 10 iterations)	1.36	1.38	1.48	1.57
SMC (1000 steps, 1 iteration)	1.10	1.48	1.50	1.69
AIS (1000 steps, 1 iteration)	1.17	1.36	1.57	1.60
SMC (1000 steps, 10 iterations)	1.39	1.39	1.41	1.51
AIS (1000 steps, 10 iterations)	1.39	1.41	1.41	1.53

schedules may be implemented (such logarithmic, quadratic) but we did not find significant improvement with such approaches.

- *Results.* Table 4.2.3 gives the average of the (unnormalized) log posterior values of the particles at time p (averaged over 10 runs), the average number of times resampling occurred for SMC and the averaged estimates of the log normalizing constant (or log marginal likelihood).

Table 4.2.3 displays the following: The particles generated by the SMC samplers have on average much higher log posterior values. The standard deviation of these values (not given here) is also significantly smaller than for AIS. However, the estimates of the normalizing constant obtained via SMC are not improved compared to AIS. For a low number of time steps p, the estimates for both algorithms are particularly poor and improve similarly as p increases. Therefore, if one is interested in estimating normalizing constants, it appears that it is preferable to use only one iterate of the kernel and more time steps. In addition, and as expected, the number of resampling steps decreases when p increases. This is because the discrepancy between consecutive densities falls, and this leads to reduced weight degeneracy. As the number of iterations per time step increases, this further reduces the number of resampling steps which we attribute to the fact that the kernels mix faster allowing us a better coverage of the space.

We now turn to Table 4.2.3 which displays estimates of the posterior means for $\{\mu_r\}$ for both algorithms. Due to the non-identifiability of the mixture components, we expect the estimated means (for each component) to be all equal and approximately 1.5. In this case, SMC provides more accurate estimates of these quantities than AIS. This is particularly significant when p is moderate ($p = 100, 200$) and when the kernel is mixing reasonably well (i.e. the number of iterations is 10). This underlines that the resampling step can improve the sampler substantially, with little extra coding effort. This is consistent with the discussion in Section 3.5.

These experimental results can also be partially explained via the expressions of the asymptotic variances (38) and (37). (We do not use multinomial resampling in our experiments and we do not resample at each iteration but the variance expressions behave similarly for more complex resampling schemes). For the estimates of the normalizing constants, when the kernel mixes perfectly (i.e. $K_k(x_{k-1}, x_k) = \pi_k(x_k)$) the terms appearing in the variance expression are of the form

$$\int \frac{\left(\widetilde{\pi}_n(x_k)\, L_{k-1}(x_k, x_{k-1})\right)^2}{\pi_{k-1}(x_{k-1})\, K_k(x_{k-1}, x_k)} \mathrm{d}x_{k-1:k} - 1 = \int \frac{\left(\pi_k(x_{k-1})\, \pi_{k+1}(x_k)\right)^2}{\pi_{k-1}(x_{k-1})\, \pi_k(x_k)} \mathrm{d}x_{k-1:k} - 1$$

when L_{k-1} is given by (30). These terms will remain high if the discrepancy between successive target distributions is large. For estimates of conditional expectations, the terms appearing in the variance expression are of the form

$$\int \frac{\left(\widetilde{\pi}_n(x_k)\, L_{k-1}(x_k, x_{k-1})\right)^2}{\pi_{k-1}(x_{k-1})\, K_k(x_{k-1}, x_k)} \left(\int \varphi(x_n)\, \widetilde{\pi}_n(x_n | x_k)\, \mathrm{d}x_n - \mathbb{E}_{\pi_n}(\varphi)\right)^2 \mathrm{d}x_{k-1:k}.$$

These terms go to zero as the mixing properties of K_k improve as in such cases $\widetilde{\pi}_n(x_n | x_k) \approx \pi_n(x_n)$.

4.2.4. Summary

In this example we have provided a comparison of SMC and AIS. For normalizing constants, SMC does not seem to improve estimation over AIS. However, for posterior expectations, it can provide a substantial gain when p is moderate and the kernels mix well. This is of importance in more complicated applications. For example, in many modern statistics problems (e.g. the population genetics example in Jasra *et al.* (2005a)), the computational cost of applying many iterations of an MCMC kernel (and thus good performance of AIS) is prohibitive and thus the usage of the resampling step can enhance the performance of the algorithm.

In the situations for which the kernels mix quickly but p is small (i.e. where SMC outperforms AIS for the same N) we might improve AIS by reducing N and increasing p to obtain similar computational cost and performance. The drawback of this approach is that it often takes a significant amount of investigation to determine an appropriate trade-off between N and p for satisfactory results; that is, SMC is often easier to calibrate (specify simulation parameters) than AIS.

For more complex problems, say if $r \geq 5$, it is unlikely that SMC will explore all of the $r!$ modes for a reasonable number of particles. However, in such contexts, the method could provide a good indication of the properties of the target density and could be used as an exploratory technique.

5. Sequential Bayesian Estimation

In the following example we present an application of SMC samplers to a sequential, trans-dimensional inference problem. In particular, we demonstrate our methodology in a case where the supports of the target distributions are nested; i.e. $E_{n-1} \subset E_n$. Such scenarios are not pathological and appear, for example, in numerous algorithms developed for counting problems in theoretical computer science; e.g. Jerrum and Sinclair (1996).

5.1. Model

We consider the Bayesian estimation of the rate of an inhomogeneous Poisson process, sequentially in time. In the static case, a similar problem was addressed in Green (1995). In the sequential case, related problems are discussed in Chopin (2004b), Fearnhead and Clifford (2003), Godsill and Vermaak (2005) and Maskell (2004).

We suppose that we record data $y_1, \ldots, y_{c_n}$ up to some time t_n with associated likelihood:

$$l_n(y_{1:c_n}|\{\lambda(u)\}_{u\leq t_n}) \propto \left[\prod_{j=1}^{c_n} \lambda(y_j)\right] \exp\left\{-\int_0^{t_n} \lambda(u)du\right\}.$$

In order to model the intensity function, we follow Green (1995) and adopt a piecewise constant function, defined for $u \leq t_n$:

$$\lambda(u) = \sum_{j=0}^{k} \lambda_j \mathbb{I}_{[\tau_j, \tau_{j+1})}(u)$$

where $\tau_0 = 0$, $\tau_{k+1} = t_n$ and the changepoints (or knots) $\tau_{1:k}$ of the regression function follow a Poisson process of intensity ν whereas for any $k > 0$

$$f(\lambda_{0:k}) = f(\lambda_0) \prod_{j=1}^{k} f(\lambda_j|\lambda_{j-1})$$

with $\lambda_0 \sim \mathcal{G}a(\mu, \upsilon)$ and $\lambda_j|\lambda_{j-1} \sim \mathcal{G}a(\lambda_{j-1}^2/\chi, \lambda_{j-1}/\chi)$.

At time t_n we restrict ourselves to the estimation of $\lambda(u)$ over the interval $[0, t_n)$. Over this interval the prior on the number k of changepoints follows a Poisson distribution of parameter νt_n

$$f_n(k) = e^{-\nu t_n} \frac{(\nu t_n)^k}{k!}$$

and, conditional on k, we have

$$f_n(\tau_{1:k}) = \frac{k!}{(t_n)^k} \mathbb{I}_{\Theta_{n,k}}(\tau_1, \dots, \tau_k)$$

where $\Theta_{n,k} = \{\tau_{1:k} : 0 < \tau_1 < \cdots < \tau_k < t_n\}$. Thus at time t_n we have the density

$$\pi_n(\lambda_{0:k}, \tau_{1:k}, k) \propto l_n(y_{1:c_n}|\{\lambda(u)\}_{u \le t_n}) f(\lambda_0) \left[\prod_{j=1}^{k} f(\lambda_j|\lambda_{j-1})\right] f_n(\tau_{1:k}) f_n(k).$$

5.2. SMC Sampler

We will consider a sequence of strictly increasing times $\{t_n\}$. For the problem considered above, we have defined a sequence of distributions on spaces:

$$E_n = \bigcup_{k \in \mathbb{N}_0} \left(\{k\} \times (\mathbb{R}^+)^{k+1} \times \Theta_{n,k}\right).$$

That is, our densities are defined on a sequence of *nested* trans-dimensional spaces; i.e. $E_{n-1} \subset E_n$. As noted in Section 3.3.2, previously developed methodologies such as AIS and RM cannot be applied in such scenarios. Additionally, we must be careful, as in Green (1995), to construct incremental weights which are indeed well-defined Radon-Nikodym derivatives.

As noted in the trans-dimensional MCMC and SMC literatures (e.g. Green (2003), Carpenter *et al.* (1999), Doucet *et al.* (2000), Pitt and Shephard (1999)) and in Section 3.3.2, a potentially good way to generate proposals in new dimensional spaces is to use the full conditional density. We will use a similar idea to generate the new changepoints.

5.2.1. *Extend Move*

In the extend move, we modify the location of the last changepoint; that is, use the Markov kernel

$$K_n(x, \mathrm{d}x') = \delta_{\tau_{1:k-1}, \lambda_{0:k}, k}(d(\tau'_{1:k-1}, \lambda'_{0:k}, k'))\pi_n(\mathrm{d}\tau'_k|\tau_{1:k-1}, \lambda_{0:k}, k).$$

The backward kernel (25) is used.

In the context of the present problem, the full conditional density is given by

$$\pi_n(\tau'_k|\tau_{1:k-1}, \lambda_{0:k}, k) \propto \lambda_{k-1}^{n_{[\tau_{k-1}:\tau'_k)}} \lambda_k^{n_{[\tau'_k:t_n)}} \exp\left\{-\tau'_k(\lambda_{k-1} - \lambda_k)\right\} \mathbb{I}_{[\tau_{k-1}, t_n)}(\tau'_k)$$

where $n_{[a,b)} = \sum_{j=1}^{c_n} \mathbb{I}_{[a,b)}(y_j)$. It is possible to sample exactly from this distribution through composition. It is also possible to compute in closed-form its normalizing constant, which is required for the incremental weight (26).

5.2.2. Birth Move

We also adopt a birth move which is simulated as follows. We generate a new changepoint τ'_{k+1} from a uniform distribution on $[\tau_k, t_n)$ and conditional on this generate a new intensity according to its full conditional:

$$\pi_n\left(\lambda'_{k+1}|\tau'_{k+1},\lambda_k\right) \propto \left(\lambda'_{k+1}\right)^{n_{[\tau'_{k+1}:t_n)}+\lambda_k^2/\chi-1}\exp\left\{-\lambda'_{k+1}[(t_n-\tau'_{k+1})+\lambda_k/\chi]\right\}$$

all other parameters are kept the same. This leads to incremental weight:

$$\frac{\pi_n(k+1,\tau'_{1:k+1},\lambda'_{0:k+1})(t_n-\tau_k)}{\pi_{n-1}(k,\tau_{1:k+1},\lambda_{0:k+1})\pi_n\left(\lambda'_{k+1}|\tau'_{k+1},\lambda_k\right)}.$$

5.2.3. The Sampler

We thus adopt the following SMC sampler:

(a) At time n make a random choice between the extend move (chosen with probability $\alpha_n(x)$) or birth move. There is clearly no extend move possible if $k=0$.
(b) Perform selected move.
(c) Choose whether or not to resample and do so.
(d) Perform an MCMC sweep of the moves described in Green (1995). That is, we retain the same target density and thus the incremental weight is 1, due to the invariance of the MCMC kernel.

5.3. Illustration

To illustrate the approach outlined above we use the popular coal mining disaster data set analyzed in (among others) Green (1995). The data consists of the times of coal mining disasters in the UK, between 1851 and 1962. We assume inference is of interest on an annual basis and so we define 112 densities (i.e. the n^{th} density is defined up to time $t_n = n$). For illustration we take prior parameters as $\mu = 4.5$, $\upsilon = 1.5$, $\chi = 0.1$, and $\nu = 20/112$. For this example, the extend move performed better than the birth move thus we let $\alpha_n(x) = 1$ if $k \geq 1$ and 0 otherwise. The backward probability is taken as equal to $\alpha_n(x)$ when $k \geq 1$ (as this is the only state it is evaluated in).

We ran our SMC sampler with 10000 particles and resampling threshold 3000 particles, using the systematic resampling approach. The initial (importance) distribution was the prior. The C^{++} code and the data are available at the following address
`http://www.cs.ubc.ca /~arnaud/smcsamplers.html`.

Figure 1 (a) demonstrates the performance of our algorithm with respect to weight degeneracy. Here we see that after the initial difficulty of the sampler (due to the initialization from the prior, and the targets' dynamic nature - we found that using more MCMC sweeps did not improve performance) the ESS never drops below 25% of its previous value. Additionally, we resample, on average, every 8.33 time steps. The former statements are not meaningless when using resampling. This is because we found, for less efficient forward and backward kernels, that the ESS would drop to 1 or 2 if consecutive densities had regions of high probability mass in different areas of the support. Thus the plot indicates that we can indeed extend the space in an efficient manner.

Figure 1 (b) shows the intensity function for the final density (full line) in the sequence, the filtered density at each time point (i.e. $\mathbb{E}[\lambda(t_n)|y_{1:c_n}]$, the crosses) and the smoothed estimate, up to lag 10 ($\mathbb{E}[\lambda(t_n)|y_{1:c_{n+10}}]$, the pluses). We can see, as expected, that the smoothed intensity approaches the final density, with the filtered intensity displaying more variability. We found that the final rate was exactly the same as Green's (1995) trans-dimensional MCMC sampler for our target density.

The bottom row of Figure 1 illustrates the performance when we only allow the MCMC steps to operate on the last five knot points. This will reduce the amount of CPU time devoted to sampling the particles and allow us to consider a truly realistic on-line implementation. This is of interest for large datasets. Here, we see (in (c)) a similar number of resampling steps to (a). In Figure 1 (d), we observe that the estimate of the intensity function suffers (slightly), with a more elongated structure at later times (in comparison to Figure 1 (b)), reflecting the fact that we cannot update the values of early knots in light of new data.

5.4. Summary

In this example we have presented the application of SMC samplers to a trans-dimensional, sequential inference problem in Bayesian statistics. We successfully applied our methodology to the coal mining disaster data set.

One point of interest, is the performance of the algorithm if we are unable to use the backward kernel (25) in the extend step for alternative likelihood functions. We found that not performing the integration and using the approximation idea (28)-(29) could still lead to good performance; we believe that this idea may also be useful for alternative problems such as optimal filtering for nonlinear non-Gaussian state-space models.

6. Conclusion

SMC algorithms are a class of flexible and general methods to sample from distributions and estimate their normalizing constants. Simulations demonstrate that this set of methods is potentially powerful. However, the performance of these methods are highly dependent on the sequence of targets $\{\pi_n\}$, forward kernels $\{K_n\}$ and backward kernels $\{L_n\}$.

In cases where we want to use SMC to sample from a fixed target π, it would be interesting - in the spirit of path sampling (Gelman and Meng, 1998) - to obtain the optimal path (in the sense of minimizing the variance of the importance weights) for moving from an easy to sample distribution π_1 to $\pi_p = \pi$. This is a very difficult problem. Given a parametrized family $\left\{\pi_{\theta(t)}\right\}_{t\in[0,1]}$ such that $\pi_{\theta(0)}$ is easy to sample and $\pi_{\theta(1)} = \pi$, a more practical approach consists of monitoring the ESS to move adaptively on the path $\theta\left(t\right)$; see Johansen *et al.* (2005) for details.

Finally, we have restricted ourselves here to Markov kernels $\{K_n\}$ to sample the particles. However, it is possible to design kernels whose parameters are a function of the whole set of current particles as suggested in Crisan and Doucet (2000), Cappé *et al.* (2004), Chopin (2002) or West (1993). This allows the algorithm to automatically scale a proposal distribution. This idea is developed in Jasra *et al.* (2005a).

7. Acknowledgments

The authors are grateful to their colleagues and the reviewers for their comments which allowed us to significantly improve this paper. The second author is grateful to the EPSRC and the Institute of Statistical Mathematics, Tokyo, Japan for their support. The third author thanks Dave Stephens and Chris Holmes for funding and advice related to this paper.

Appendix

Proof of Proposition 3.1. The result follows easily from the variance decomposition formula

$$\operatorname{var}[w_n(X_{1:n})] = \mathbb{E}\big(\operatorname{var}[w_n(X_{1:n})|X_n]\big) + \operatorname{var}\big(\mathbb{E}[w_n(X_{1:n})|X_n]\big). \tag{42}$$

The second term on the right hand side of (42) is independent of the backward Markov kernels $\{L_k\}$ as

$$\mathbb{E}\left[\left. w_n\left(X_{1:n}\right)\right| X_n\right] = \frac{\gamma_n\left(X_n\right)}{\eta_n\left(X_n\right)}$$

whereas $\operatorname{var}\left[\left. w\left(X_{1:n}\right)\right| X_n\right]$ is equal to zero if one uses (21).

Proof of Proposition 3.2. The expression (35) follows from the delta method. Expression (37) follows from a convenient rewriting of the variance expression established in (Del Moral, 2004; Proposition 9.4.2, pp. 302); see also (Chopin, 2004; Theorem 1) for an alternative derivation. The variance is given by

$$\sigma^2_{SMC,n}\left(\varphi\right) = \mathbb{E}_{\eta_1}\left[\overline{w}_1^2 Q_{2:n}\left(\varphi - \mathbb{E}_{\pi_n}\left(\varphi\right)\right)^2\right] + \sum_{k=2}^{n} \mathbb{E}_{\pi_{k-1}K_k}\left[\overline{w}_k^2 Q_{k+1:n}\left(\varphi - \mathbb{E}_{\pi_n}\left(\varphi\right)\right)^2\right] \tag{43}$$

where the semigroup, Q, is defined as $Q_{n+1:n}\left(\varphi\right) = \varphi$,

$$Q_{k+1:n}\left(\varphi\right) = Q_{k+1} \circ \cdots \circ Q_n\left(\varphi\right)$$

and

$$Q_n\left(\varphi\right)\left(x_{n-1}\right) = \mathbb{E}_{K_n\left(x_{n-1},\cdot\right)}\left[\overline{w}_n\left(x_{n-1}, X_n\right)\varphi\left(X_n\right)\right]$$

where

$$\begin{aligned}\overline{w}_n\left(x_{n-1}, x_n\right) &= \frac{\pi_n\left(x_n\right) L_{n-1}\left(x_n, x_{n-1}\right)}{\pi_{n-1}\left(x_{n-1}\right) K_n\left(x_{n-1}, x_n\right)}\\ &= \frac{Z_{n-1}}{Z_n}\widetilde{w}_n\left(x_{n-1}, x_n\right).\end{aligned}$$

The expression (43) is difficult to interpret. It is conveniently rearranged here. The key is to notice that

$$\begin{aligned}Q_n\left(\varphi\right)\left(x_{n-1}\right) &= \mathbb{E}_{K_n\left(x_{n-1},\cdot\right)}\left[\overline{w}_n\left(x_{n-1}, X_n\right)\varphi\left(X_n\right)\right]\\ &= \int K_n\left(x_{n-1}, x_n\right)\frac{\pi_n\left(x_n\right) L_{n-1}\left(x_n, x_{n-1}\right)}{\pi_{n-1}\left(x_{n-1}\right) K_n\left(x_{n-1}, x_n\right)}\varphi\left(x_n\right) \mathrm{d}x_n\\ &= \frac{1}{\pi_{n-1}\left(x_{n-1}\right)}\int \varphi\left(x_n\right)\pi_n\left(x_n\right) L_{n-1}\left(x_n, x_{n-1}\right) \mathrm{d}x_n.\\ &= \frac{\widetilde{\pi}_n\left(x_{n-1}\right)}{\pi_{n-1}\left(x_{n-1}\right)}\int \varphi\left(x_n\right)\widetilde{\pi}_n\left(\left. x_n\right| x_{n-1}\right) \mathrm{d}x_n\end{aligned}$$

Similarly, one obtains

$$\begin{aligned}
&Q_{n-1:n}\left(\varphi\right)\\
&=Q_{n-1}\left(Q_{n}\left(\varphi\right)\right)\left(x_{n-2}\right)\\
&=\mathbb{E}_{K_{n-1}(x_{n-2},\cdot)}\left[w_{n-1}\left(x_{n-2:n-1}\right)Q_{n}\left(\varphi\right)\left(x_{n-1}\right)\right]\\
&=\frac{1}{\pi_{n-2}\left(x_{n-2}\right)}\int\left(\frac{1}{\pi_{n-1}\left(x_{n-1}\right)}\int\varphi\left(x_{n}\right)\pi_{n}\left(x_{n}\right)L_{n-1}\left(x_{n},x_{n-1}\right)\mathrm{d}x_{n}\right)\\
&\qquad\qquad\times\pi_{n-1}\left(x_{n-1}\right)L_{n-2}\left(x_{n-1},x_{n-2}\right)\mathrm{d}x_{n-1}.\\
&=\frac{1}{\pi_{n-2}\left(x_{n-2}\right)}\int\left(\int\varphi\left(x_{n}\right)\widetilde{\pi}_{n}\left(\left.x_{n-1:n}\right|x_{n-2}\right)\mathrm{d}x_{n-1:n}\right)\widetilde{\pi}_{n-2}\left(x_{n-2}\right)\mathrm{d}x_{n-1}.\\
&=\frac{\widetilde{\pi}_{n-1}\left(x_{n-2}\right)}{\pi_{n-2}\left(x_{n-2}\right)}\int\varphi\left(x_{n}\right)\widetilde{\pi}_{n}\left(\left.x_{n}\right|x_{n-2}\right)\mathrm{d}x_{n}
\end{aligned}$$

and, by induction, one gets

$$\begin{aligned}
Q_{k+1:n}\left(\varphi\right)&=\frac{1}{\pi_{k}\left(x_{k}\right)}\int\cdots\int\varphi\left(x_{n}\right)\pi_{n}\left(x_{n}\right)\prod_{i=k}^{n-1}L_{i}\left(x_{i},x_{i-1}\right)\mathrm{d}x_{k+1:n}. \qquad (44)\\
&=\frac{\widetilde{\pi}_{n}\left(x_{k}\right)}{\pi_{k}\left(x_{k}\right)}\int\varphi\left(x_{n}\right)\widetilde{\pi}_{n}\left(\left.x_{n}\right|x_{k}\right)\mathrm{d}x_{n}.
\end{aligned}$$

The expression of $\sigma^2_{SMC,n}\left(\varphi\right)$ given (37) follows now directly from (44) and (43). Similarly we can rewrite the variance expression established in (Del Moral, 2004; Proposition 9.4.1, pp. 301) and use the delta method to establish (38).

References

Cappé, O., Guillin, A., Marin, J. M. and Robert, C. P. (2004) Population Monte Carlo. *J. Comp. Graph. Statist.*, **13**, 907–930.

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

Celeux, G., Marin, J. M. and Robert, C. P. (2006) Iterated importance sampling in missing data problems. *Comp. Statist. Data Anal.*

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

Chopin, N. (2004a) Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Ann. Statist.*, **32**, 2385–2411.

Chopin, N. (2004b) Dynamic detection of changepoints in long time series. Technical report, University of Bristol.

Crisan, D. and Doucet, A. (2000) Convergence of sequential Monte Carlo methods. Technical report, University of Cambridge, CUED/F-INFENG/TR381.

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

Del Moral, P. and Doucet, A. (2003) On a class of genealogical and interacting metropolis models. In *Séminaire de Probabilités XXXVII* (Eds J. Azéma, M. Emery, M. Ledoux and M. Yor), pp. 415–446. Berlin: Springer.

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

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

Fearnhead, P. and Clifford, P. (2003) On-line inference for well-log data. *J. R. Statist. Soc. B*, **65**, 887–899.

Gelman, A. and Meng, X. L. (1998) Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statist. Sci.*, **13**, 163–185.

Gilks, W. R. and Berzuini, C. (2001) Following a moving target - Monte Carlo inference for dynamic Bayesian models. *J. R. Statist. Soc. B*, **63**, 127–146.

Givens, G. H. and Raftery, A. E. (1996) Local adaptive importance sampling for multivariate densities with strong non-linear relationships. *J. Amer. Statist. Assoc.*, **91**, 132–141.

Godsill, S. J. and Vermaak, J. (2005) Variable rate particle filters for tracking applications. *Proc. Workshop IEEE Stat. Sig Proc.*

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

Green, P. J. (2003) Trans-dimensional Markov chain Monte Carlo. In *Highly Structured Stochastic Systems* (Eds P. J. Green, N. L. Hjort and S. Richardson), pp. 179–196. Oxford: Oxford University Press.

Jarzynski, C. (1997) Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.*, **78**, 2690–2693.

Jasra, A., Doucet, A., Stephens, D. A. and Holmes, C. C. (2005a) Interacting sequential Monte Carlo samplers for trans-dimensional simulation. Technical report, Imperial College London.

Jasra, A., Holmes, C. C. and Stephens, D. A. (2005b) Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modelling. *Statist. Sci.*, **20**, 50–67.

Jerrum, M. and Sinclair, A. (1996) The Markov chain Monte Carlo method: An approach to approximate counting and integration. In *Approximation algorithms for NP hard problems* (Ed. D. S. Hoschbaum), pp. 482–520. Boston: PWS publishing.

Johansen, A., Del Moral, P. and Doucet, A. (2005) Sequential Monte Carlo samplers for rare event estimation. Technical report, University of Cambridge, CUED/F-INFENG/543.

Kitagawa, G. (1996) Monte Carlo filter and smoother for non-gaussian, non-linear state space models. *J. Comp. Graph. Statist.*, **5**, 1–25.

Künsch, H. R. (2005) Recursive Monte Carlo filters: algorithms and theoretical analysis. *Ann. Statist.*, **33**, 1983–2021.

Liang, F. (2002) Dynamically weighted importance sampling in Monte Carlo computation. *J. Amer. Statist. Assoc.*, **97**, 807–821.

Liu, J. and Chen, R. (1998) Sequential Monte Carlo for dynamic systems. *J. Amer. Statist. Assoc.*, **93**, 1032–1044.

Liu, J. S. (2001) *Monte Carlo Strategies in Scientific Computing.* New York: Springer.

MacEachern, S. N., Clyde, M. and Liu, J. S. (1999) Sequential importance sampling for nonparametric Bayes: the next generation. *Cand. J. Statist.*, **27**, 251–267.

Maskell, S. (2004) Joint tracking of manoeuvring targets and classification of their manoeuvrability. *J. Appl. Sig. Proc.*, **15**, 2239–2350.

Neal, R. (2001) Annealed importance sampling. *Statist. Comp.*, **11**, 125–139.

Oh, M. S. and Berger, J. (1993) Integration of multimodal functions by monte carlo importance sampling. *J. Amer. Statist. Assoc.*, **88**, 450–456.

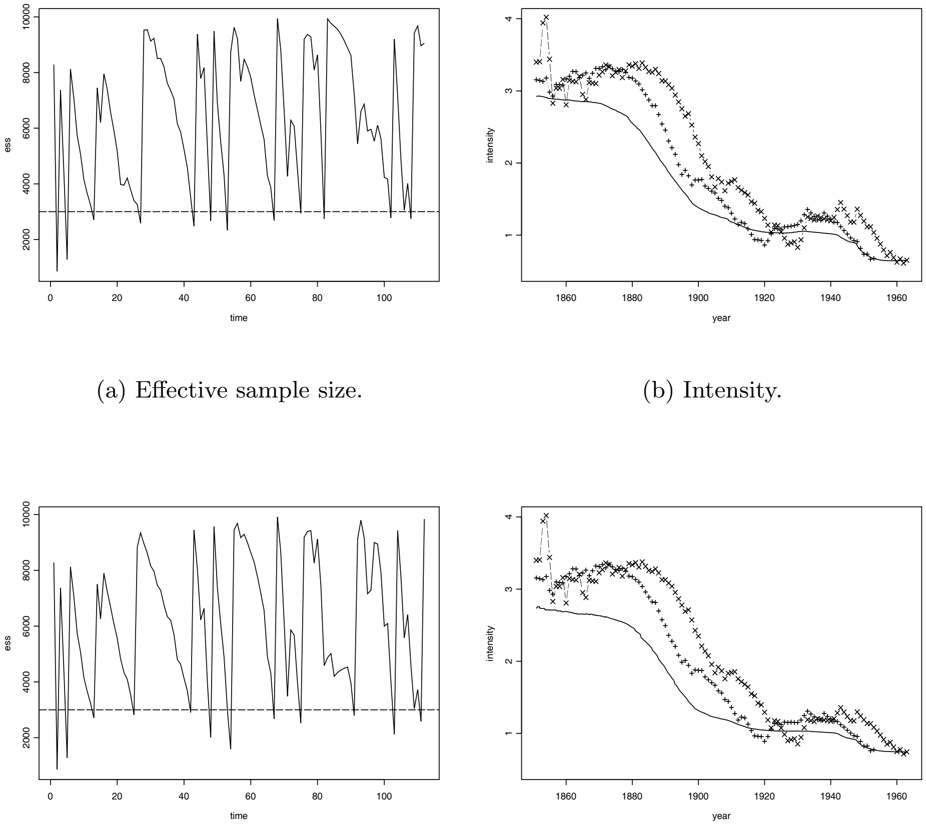
Pitt, M. K. and Shephard, N. (1999) Filtering via simulation: auxiliary particle filter. *J. Amer. Statist. Assoc.*, **94**, 590–599.

Richardson, S. and Green, P. J. (1997) On Bayesian analysis of mixture models with an unknown number of components (with discussion). *J. R. Statist. Soc. B*, **59**, 731–792.

Robert, C. P. and Casella, G. (2004) *Monte Carlo Statistical Methods.* New York: Springer, second edition.

Rousset, M. and Stoltz, G. (2005) Equilibrium sampling from non-equilibrium dynamics. Technical report, Université Nice Sophia Antipolis.

West, M. (1993) Approximating posterior distributions by mixture. *J. R. Statist. Soc. B*, **55**, 409–422.



(a) Effective sample size.

(b) Intensity.

(c) Effective sample size.

(d) Intensity.

Fig. 1. Effective sample size plot and intensity function for Coal mining disaster data. We ran 10000 particles for 112 densities and resampling threshold ($(--)$ in (a) & (c)) 3000 particles. In the intensity plots, the full-line is the estimated intensity given the entire data, the crosses, the filtered density at each time and the pluses the smoothed estimate (lag 10). The top row ((a) & (b)) are the results when the MCMC steps operate upon the entire state-space the bottom row ((c) & (d)) are the results when the MCMC steps only operate upon the last 5, or fewer, knot points.