Sequential Monte Carlo Samplers

Pierre Del Moral

CNRS-UMR C55830 and University Paul Sabatier, Toulouse, France.

Arnaud Doucet†, Gareth W. Peters

Cambridge University, UK.

Summary. In this paper, we propose a general 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 not only to derive simple algorithms to make parallel Markov chain Monte Carlo runs interact in a principled way, but also to obtain new methods for global optimization and sequential Bayesian estimation. We demonstrate the performance of these algorithms through simulation for various integration and global optimization tasks arising in the context of Bayesian inference.

Keywords: Genetic Algorithm, Importance Sampling, Resampling, Markov chain Monte Carlo, Sequential Monte Carlo, Simulated Annealing.

1. Introduction

Consider a sequence of probability distributions $\{\pi_n\}_{n\in\mathcal{N}}$ defined on a common measurable space E where $\mathcal{N} = \{1, \ldots, p\}$ or $\mathcal{N} = \mathbb{N}^+$. For ease of presentation we will assume that each distribution $\pi_n(dx)$ admits a probability density $\pi_n(x)$ with respect to a dominating measure denoted dx. We will also refer to n as the time index; this variable is simply a counter and need not have any relation with "real time". In this paper, we are interested in sampling this sequence of distributions *sequentially*; that is to first sample from π_1 then from π_2 and so on.

This problem arises in numerous applications. One use of sequential methods is to move from a tractable (easy to sample) distribution π_1 to a distribution of interest, say π_p, through a sequence of artificial intermediate distributions (Neal, 2001). In the context of sequential Bayesian inference, π_n could be the posterior distribution of a parameter given the data collected till time n; say $\pi_n(x) = p(x|\, y_1, \ldots, y_n)$. In a batch setup where a fixed set of observations $y_1, \ldots, y_T$ is available, one could also consider the sequence of distributions $p(x|\, y_1, \ldots, y_n)$ for $n \leq T$ for the following two reasons. First, for huge datasets, standard simulation methods such as Markov Chain Monte Carlo (MCMC) methods require a complete "browsing" of the observations, by 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, in the context of optimization, and similarly to simulated annealing, one could consider the

†*Address for correspondence:* Arnaud Doucet, Information Engineering Division, Department of Engineering, Cambridge University, Trumpington Street, CB2 1PZ Cambridge, UK. Email: ad2@eng.cam.ac.uk

sequence of distributions $\pi_n(x) \propto [\pi(x)]^{\gamma_n}$ for an increasing schedule $\{\gamma_n\}_{n\in\mathcal{N}}$. Finally, one could simply consider the case where $\pi_n = \pi$ for all $n \in \mathcal{N}$.

The tools favoured by statisticians to sample from complex distributions are MCMC methods; see for example (Robert and Casella, 1999). 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, Gibbs steps etc. MCMC algorithms have been successfully used in many applications in statistics. Two problems with MCMC are that it is difficult to assess when the Markov chain has reached its stationary regime and it can easily get stuck in local modes. Moreover, MCMC cannot be used in a sequential Bayesian estimation context.

We propose here a different approach to sample from $\{\pi_n\}_{n\in\mathcal{N}}$ based on 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 be ingredients of the methods proposed here in most cases. SMC methods have been recently studied and used extensively in the context of sequential Bayesian inference and physics. At a given time n, the basic idea is to obtain a large collection of N ($N \gg 1$) 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 $\varphi : E \to \mathbb{R}$

$$\sum_{i=1}^{N} W_n^{(i)} \varphi\left(X_n^{(i)}\right) \underset{N\to\infty}{\to} E_{\pi_n}(\varphi)$$

where

$$E_{\pi_n}(\varphi) \triangleq \int \varphi(x)\, \pi_n(x)\, dx. \tag{1}$$

These particles are carried forward over time using a combination of Sequential Importance Sampling (SIS) and resampling ideas. This approach is completely different from parallel MCMC/tempering mechanisms where one runs an MCMC chain on an extended space E^N. In those cases, one specifies a *joint* invariant distribution on E^N for the particles (Geyer and Thompson, 1995) whereas the use of SMC samplers requires only the specification of a distribution on E.

Standard SMC algorithms available in the literature do not apply to our problem. This is because these algorithms deal with the case whereby the target distribution of interest at time n is defined on E_n with $\dim(E_{n-1}) < \dim(E_n)$; e.g. $E_n = E^n$ whereas we are interested in the case where the distributions $\{\pi_n\}_{n\in\mathcal{N}}$ are all defined on a common space E. To be able to use the SMC methodology, we build an artificial sequence of distributions $\{\widetilde{\pi}_n\}_{n\in\mathcal{N}}$ defined on $E_n = E^n$ with $\widetilde{\pi}_n$ admitting a marginal π_n. Our approach has some connections with Annealed Importance Sampling (AIS) (Neal, 2001) and the algorithms recently proposed in (Chopin, 2002) and (Cappé *et al.*, 2004) which are detailed in Section 2. However, the generic framework we present here is more general and allows us to develop new algorithms to make parallel MCMC runs interact in a simple and principled way, to perform global optimization, solve sequential Bayesian estimation problems or compute the probabilities of rare events. As for MCMC, the performance of these algorithms is highly dependent on the target distributions $\{\widetilde{\pi}_n\}_{n\in\mathcal{N}}$ and proposal distributions used to explore the space. Guidelines for the design of efficient algorithms are given and illustrated by examples.

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 work we provide the expression for the asymptotic variance of the resulting estimates.

The rest of the paper is organized as follows. In Section 2, we present a brief review of a generic SMC algorithm which samples from a sequence of distributions $\{\widetilde{\pi}_n\}_{n\in\mathcal{N}}$ defined on $E_n = E^n$. We then show how one can build a sequence of distributions $\{\widetilde{\pi}_n\}_{n\in\mathcal{N}}$ which admits fixed marginals, and we provide a way to design these distributions so as to obtain efficient algorithms. Some extensions and connections with previous work are outlined. Section 3 applies this class of algorithms to Bayesian variable selection and a sequential Bayesian estimation problem. Finally, we discuss a few open methodological and theoretical problems in Section 4. The proofs of our propositions are given in the Appendix.

2. Sequential Monte Carlo Samplers

2.1. A Generic SMC Algorithm

Consider a sequence of distributions $\{\widetilde{\pi}_n\}_{n\in\mathcal{N}}$ defined on $E_n = E^n$; each distribution $\widetilde{\pi}_n(dx_{1:n})$ admits a density $\widetilde{\pi}_n(x_{1:n})$ with respect to a dominating measure denoted $dx_{1:n}$. We describe here briefly a generic SMC algorithm to sample from $\{\widetilde{\pi}_n\}_{n\in\mathcal{N}}$ based on a Sampling Importance Resampling strategy; see (Doucet *et al.*, 2001) for a book-length survey of the SMC literature. Alternative SMC algorithms such as the Auxiliary Particle method of Pitt and Shephard (1999) are equally valid.

Later we will use the notation $X_{i:j}$, $i \leq j$, (resp. $x_{i:j}$) to denote $(X_i, \ldots, X_j)$ (resp. $(x_i, \ldots, x_j)$). At time $n-1$, assume a set of weighted particles $\left\{W_{n-1}^{(i)}, X_{1:n-1}^{(i)}\right\}$ $(i = 1, \ldots N$, $W_{n-1}^{(i)} > 0$, $\sum_{i=1}^{N} W_{n-1}^{(i)} = 1)$ approximating $\widetilde{\pi}_{n-1}$ is available, i.e. the empirical measure

$$\widehat{\widetilde{\pi}}_{n-1}(dx_{1:n-1}) = \sum_{i=1}^{N} W_{n-1}^{(i)} \delta_{X_{1:n-1}^{(i)}}(dx_{1:n-1}),$$

is an approximation of $\widetilde{\pi}_{n-1}$. At time n, we extend the path of each particle with a Markov‡ kernel $K_n(x, x')$ giving the probability or probability density of moving to x' when the current state is x. Importance sampling can then be used to correct for the discrepancy between the sampling distribution and $\widetilde{\pi}_n(x_{1:n})$; the normalized weights are given by

$$W_n^{(i)} \propto W_{n-1}^{(i)} w_n\left(X_{1:n}^{(i)}\right), \quad \sum_{i=1}^{N} W_n^{(i)} = 1, \tag{2}$$

where the incremental weight is equal to

$$w_n(x_{1:n}) = \frac{\widetilde{\pi}_n(x_{1:n})}{\widetilde{\pi}_{n-1}(x_{1:n-1}) K_n(x_{n-1}, x_n)}. \tag{3}$$

‡The Markov assumption could be relaxed.

Degeneracy of the particle approximation is routinely measured using the Effective Sample Size (ESS) criterion $\left(\sum_{i=1}^{N}\left(W_n^{(i)}\right)^2\right)^{-1}$ (Liu, 2001). The ESS takes values between 1 and N. If the degeneracy is too high, i.e. the ESS is below a prespecified threshold, say $N/2$, each particle $X_{1:n}^{(i)}$ is copied $N_n^{(i)}$ times under the constraint $\sum_{i=1}^{N} N_n^{(i)} = N$; $N_n^{(i)}$ being approximately proportional to $W_n^{(i)}$ 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 $\widehat{\widetilde{\pi}}_n\left(dx_{1:n}\right)$; 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), residual resampling (Liu, 2001) or minimum entropy resampling (Crisan, 2001) can also be used and reduce the variance of $N_n^{(i)}$ over the multinomial scheme. All these resampling schemes are unbiased; i.e. $E\left(N_n^{(i)} \middle| \left\{W_n^{(i)}\right\}\right) = NW_n^{(i)}$.

To summarise, the algorithm proceeds as follows. We denote the initial importance distribution as μ_1.

Initialization; $n = 1$.

Sampling step x

- For $i = 1, ..., N$, sample $X_1^{(i)} \sim \mu_1(\cdot)$.
- For $i = 1, ..., N$, evaluate the normalized weights $W_1^{(i)}$

$$W_1^{(i)} \propto \frac{\widetilde{\pi}_1\left(X_1^{(i)}\right)}{\mu_1\left(X_1^{(i)}\right)}, \quad \sum_{i=1}^{N} W_1^{(i)} = 1. \tag{4}$$

Resampling step

- If ESS<Threshold then resample particles $\left\{W_1^{(i)}, X_1^{(i)}\right\}$ to obtain N new particles $\left\{N^{-1}, X_1^{(i)}\right\}$.

At time n; $n \in \mathcal{N} \backslash \{1\}$.

Sampling step

- For $i = 1, ..., N$, sample $X_n^{(i)} \sim K_n\left(X_{n-1}^{(i)}, \cdot\right)$.
- For $i = 1, ..., N$, evaluate the normalized weights $W_n^{(i)}$ using (2) and (3).

Resampling step

- If ESS<Threshold then resample particles $\left\{W_n^{(i)}, X_n^{(i)}\right\}$ to obtain N new particles $\left\{N^{-1}, X_n^{(i)}\right\}$.

The complexity of this algorithm is in $O(N)$ and it can be parallelized easily.

This algorithm can also be used to estimate the ratio of normalizing constants. Indeed, typically the sequence of distributions $\widetilde{\pi}_n\left(x_{1:n}\right)$ is known only up to a normalizing constant,

i.e. $\widetilde{\pi}_n(x_{1:n}) = Z_n^{-1} f_n(x_{1:n})$ where $f_n(\cdot)$ can be evaluated pointwise and Z_n is unknown. In this case, the unnormalized incremental importance weights we compute are $\widetilde{w}_n\left(X_{1:n}^{(i)}\right)$ where

$$\widetilde{w}_n(x_{1:n}) = \frac{f_n(x_{1:n})}{f_{n-1}(x_{1:n-1}) K_n(x_{n-1}, x_n)}.$$

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

$$\frac{Z_n}{Z_{n-1}} = \frac{\int f_n(x_{1:n})\, dx_{1:n}}{\int f_{n-1}(x_{1:n-1})\, dx_{1:n-1}} \text{ by } \widehat{\frac{Z_n}{Z_{n-1}}} = \sum_{i=1}^{N} W_{n-1}^{(i)} \widetilde{w}_n\left(X_{1:n}^{(i)}\right). \tag{5}$$

2.2. Sequence of Joint Distributions and Asymptotic Variances

Our aim is to sample from $\{\pi_n\}_{n\in\mathcal{N}}$ where each π_n is defined on E. Up to this point we have shown how to obtain a cloud of weighted particles which approximate an arbitrary sequence of distributions $\{\widetilde{\pi}_n\}_{n\in\mathcal{N}}$ where $\widetilde{\pi}_n$ is defined on $E_n = E^n$. It follows straightforwardly that if $\widetilde{\pi}_n(x_{1:n})$ admits $\pi_n(x_n)$ as a marginal, then we will have achieved our goal. It is easy to build such a sequence $\{\widetilde{\pi}_n\}_{n\in\mathcal{N}}$. Consider $\widetilde{\pi}_1(x_1) = \pi_1(x_1)$ and for $n \geq 2$ a distribution of the form

$$\widetilde{\pi}_n(x_{1:n}) = \pi_n(x_n)\, \widetilde{\pi}_n(x_{1:n-1} | x_n) \tag{6}$$

where, for any $x_n \in E$, $\widetilde{\pi}_n(x_{1:n-1} | x_n)$ is an arbitrary probability distribution on E_{n-1}. By construction, we have $\int \widetilde{\pi}_n(x_{1:n})\, dx_{1:n-1} = \pi_n(x_n)$.

A choice for $\widetilde{\pi}_n(x_{1:n-1} | x_n)$ which allows a simple recursive evaluation of the importance weights is given by

$$\widetilde{\pi}_n(x_{1:n-1} | x_n) = \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k) \tag{7}$$

where $\{L_n\}$ is a sequence of *auxiliary* Markov transition kernels with $L_k(x, x')$ giving the probability or probability density of moving to x' when the current state is x. Indeed, in this case, it follows straightforwardly using (6)-(7) that the incremental weight (3) can be computed up to a normalizing constant easily and satisfies

$$w_n(x_{1:n}) = w_n(x_{n-1}, x_n) = \frac{\pi_n(x_n) L_{n-1}(x_n, x_{n-1})}{\pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)}. \tag{8}$$

The optimal selection of the kernels $\{L_n\}$ will be discussed in the next section. We will also show that many recent algorithms are just special cases of this framework which define the kernels $\{L_n\}$ implicitly (Cappé *et al.*, 2004; Chopin, 2002) or explicitly (Neal, 2001).

The resulting algorithm can be interpreted as an adaptive importance sampling resampling technique. The key point is that the introduction of the auxiliary kernels $\{L_n\}$ allows us to use importance sampling without having to compute analytically the marginal distribution of the particles $\left\{X_n^{(i)}\right\}$, which typically does not admit a closed-form expression.

In our context, we are not interested in estimating the joint distribution $\widetilde{\pi}_n$, but only its marginal π_n thus the memory requirements are in $O(N)$ and do not increase over time

as we only need to keep $\left\{W_n^{(i)}, X_n^{(i)}\right\}$ in memory at time n§. These particles yield the following estimate of (1)

$$\widehat{E}_{\pi_n}(\varphi) = \sum_{i=1}^{N} W_n^{(i)} \varphi\left(X_n^{(i)}\right). \tag{9}$$

If $\pi_n = \pi$ for $n \in \mathcal{N}$, then an alternative estimate can also be used

$$\widehat{E}_{\pi_n}(\varphi) = \frac{1}{n} \sum_{k=1}^{n} \sum_{i=1}^{N} W_k^{(i)} \varphi\left(X_k^{(i)}\right). \tag{10}$$

We now compute the asymptotic variance of the estimate (9) 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. The variance expression for general SMC algorithms has previously been established in the literature. However, we propose here a new representation which clarifies the influence of the kernels $\{L_n\}$.

Proposition 1. *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*

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

with

$$\sigma_{IS,n}^2(\varphi) = \int \frac{\widetilde{\pi}_n^2(x_{1:n})}{\mu_n(x_{1:n})} \left(\varphi(x_n) - E_{\pi_n}(\varphi)\right)^2 dx_{1:n} \tag{11}$$

where the importance distribution μ_n is given by

$$\mu_n(x_{1:n}) = \mu_1(x_1) \prod_{k=2}^{n} K_k(x_{k-1}, x_k).$$

When multinomial resampling is used at each iteration, one has

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

where, for $n \geq 2$,

$$\begin{aligned} & \sigma_{SMC,n}^2(\varphi) \qquad (12) \\ = & \int \frac{\widetilde{\pi}_n^2(x_1)}{\mu_1(x_1)} \left(\int \varphi(x_n)\, \widetilde{\pi}_n(x_n|x_1)\, dx_n - E_{\pi_n}(\varphi)\right)^2 dx_1 \\ & + \sum_{k=2}^{n-1} \int \frac{\left(\widetilde{\pi}_n(x_k)\, L_{k-1}(x_k, x_{k-1})\right)^2}{\pi_{k-1}(x_{k-1})\, K_k(x_{k-1}, x_k)} \left(\int \varphi(x_n)\, \widetilde{\pi}_n(x_n|x_k)\, dx_n - E_{\pi_n}(\varphi)\right)^2 dx_{k-1:k} \\ & + \int \frac{\left(\pi_n(x_n)\, L_{n-1}(x_n, x_{n-1})\right)^2}{\pi_{n-1}(x_{n-1})\, K_n(x_{n-1}, x_n)} \left(\varphi(x_n) - E_{\pi_n}(\varphi)\right)^2 dx_{n-1:n}. \end{aligned}$$

§The memory required to store particles does not increase over time but, in the case of sequential Bayesian inference, it might be necessary to store the whole sequence of observations.

In (12), we denote $\int \widetilde{\pi}_n(x_{1:n})\, dx_{1:k-1} dx_{k:n}$ by $\widetilde{\pi}_n(x_k)$ and $\int \widetilde{\pi}_n(x_{1:n})\, dx_{1:k-1} dx_{k:n-1} / \widetilde{\pi}_n(x_k)$ by $\widetilde{\pi}_n(x_n | x_k)$. In the general case, we cannot claim that $\sigma^2_{SMC,n}(\varphi) < \sigma^2_{IS,n}(\varphi)$. In particular, if the importance weights do not have a large variance, resampling is typically wasteful. In practice, as mentioned ealier, we only resample when this variance (measured through the ESS) is high. In these cases, resampling typically helps as it has the effect of somehow resetting the system.

Expression (12) also shows the impact of the sequence of kernels $\{L_n\}$ on the performance of the algorithm. The faster the sequence of kernels $\{L_n\}$ mixes, the faster $\widetilde{\pi}_n(x_n | x_k)$ converges to $\pi_n(x_n)$ as $n-k$ increases. It would be tempting to select $L_k(x, x') = \pi_k(x')$ for any k as then $\widetilde{\pi}_n(x_n | x_k) = \pi_n(x_n)$ and (12) simplifies to

$$\sigma^2_{SMC,n}(\varphi) = \int \frac{\pi_n^2(x_n)\, \pi_{n-1}(x_{n-1})}{K_n(x_{n-1}, x_n)} \left(\varphi(x_n) - E_{\pi_n}(\varphi)\right)^2 dx_{n-1:n}.$$

However, loosely speaking, to ensure that the variance is small for any function φ, we need to control the importance weight

$$\frac{\pi_n(x_n)}{K_n(x_{n-1}, x_n)}$$

over $E \times E$. In most cases, this rules out the use of MCMC moves as this ratio is not defined; e.g. if $\pi_n(x_n)$ is a probability density on $\mathbb{R}$ with respect to the Lebesgue measure and K_n a Metropolis-Hastings kernel. Now, if $L_k(x, x') \neq \pi_{k-1}(x')$, then a sum of terms appears in the expression of the variance, but weights of the form

$$\frac{\widetilde{\pi}_n(x_k)\, L_{k-1}(x_k, x_{k-1})}{\pi_{k-1}(x_{k-1})\, K_k(x_{k-1}, x_k)}$$

can be controlled more easily if L_k is chosen conveniently as a function of π_{k-1} and K_k. We will discuss in the next subsection how the kernels $\{L_n\}$ could be chosen.

2.3. Algorithm Settings

The algorithm presented in the previous subsection is very general. There are many potential choices for $\{\pi_n\}_{n \in \mathcal{N}}$ leading to various integration and optimization algorithms.

Sequence of distributions $\{\pi_n\}_{n \in \mathcal{N}}$.

• As suggested by Neal (2001), it can be of interest to consider an inhomogeneous sequence of distributions which 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 & Meng, 1998)

$$\pi_n(x) \propto [\pi(x)]^{\gamma_n} [\mu_1(x)]^{1-\gamma_n} \tag{13}$$

with $\mathcal{N} = \{1, \ldots, p\}$, $\gamma_1 = 0$, $\gamma_n > \gamma_{n-1}$ for $n = 2, \ldots, p$ and $\gamma_p = 1$.

• In the context of Bayesian inference for static parameters where T observations $(y_1, \ldots, y_T)$ are available, one can consider

$$\pi_n(x) = p(x | y_{1:n}). \tag{14}$$

with $\mathcal{N} = \{1, \ldots, T\}$.

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

$$\pi_n(x) \propto [\pi(x)]^{\gamma_n} \tag{15}$$

where $\mathcal{N} = \mathbb{N}^+$, $\{\gamma_n\}_{n\geq 1}$ is an increasing sequence such that $\gamma_n \to \infty$. In this case, the resulting algorithm is a genetic algorithm where the sampling step is the "mutation" step and the resampling step is the selection step (Goldberg, 1989). However, there is a significant difference to standard genetic algorithms as we control the asymptotic ($N \to \infty$) distribution of the particles. This makes the analysis of the resulting algorithm easier. Indeed, standard selection mutation genetic algorithms proposed in the literature are based on

$$X_n^{(i)} \sim K_n\left(X_{n-1}^{(i)}, \cdot\right) \text{ and } W_n^{(i)} \propto \pi_n\left(X_n^{(i)}\right).$$

This corresponds to sampling from a joint distribution given by

$$\widetilde{\pi}_n(x_{1:n}) \propto \mu_1(x_1)\,\pi_1(x_1) \prod_{k=2}^{n} K_k(x_{k-1}, x_k)\,\pi_k(x_k).$$

In the general case, $\widetilde{\pi}_n(x_n) \neq \pi_n(x_n)$ and the particles might not concentrate themselves on the set of global maxima of π.

• Consider the case where we are interested in estimating the probability of a very rare event, A, under the distribution π ($\pi(A) \ll 1$). This has numerous applications in finance and telecommunications. In most of these applications, π is typically easy to sample from and its normalizing constant is known. We can consider the sequence of distributions

$$\pi_n(x) \propto \pi(x)\,\mathbf{1}_{A_n}(x)$$

with $\mathcal{N} = \{1, \ldots, p\}$, where $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_{p-1} \supseteq A_p$, $A_1 = E$ and $A_p = A$. Using (5), an estimate of $\pi(A)$ is given by

$$\widehat{\pi}(A) = Z_1 \prod_{k=1}^{p-1} \widehat{\frac{Z_{k+1}}{Z_k}}.$$

• One can simply consider the case where $\pi_n = \pi$ for all $n \in \mathcal{N}$ but we shall explain why we do not think this is the most useful case in practice.

Sequence of auxiliary kernels $\{L_n\}_{n\in\mathcal{N}}$.

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 completely arbitrary. However, in practice $\{L_n\}$ should be optimized with respect to $\{K_n\}$ in order to obtain good performance. This clearly appears in (12). Instead of directly minimizing $\sigma^2_{SMC,n}(\varphi)$ with respect to $\{L_n\}$, we concentrate on the variance of the importance weights - which is independent of φ.

For any probability density η, we use the following notation

$$\eta K_{i:j}(x_j) \triangleq \int \eta(x_{i-1}) \prod_{k=i}^{j} K_k(x_{k-1}, x_k)\, dx_{i-1:j-1}.$$

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

$$\mu_n(x_n) = \mu_1 K_{2:n}(x_n) \tag{16}$$

if the particles have not been resampled before time n and

$$\mu_n(x_n) = \pi_l K_{l+1:n}(x_n) \tag{17}$$

if the last time the particles were resampled was l. For sake of simplicity, we consider here the case (16), note that the more general case (17) can be handled similarly.

To understand how to select $\{L_n\}$, it is worth remembering that we are interested in sampling from π_n and we have particles distributed at time n according to μ_n. If we could compute (17) analytically, we would simply correct for the discrepancy between π_n and μ_n by computing the (unnormalized) importance weight

$$\frac{\pi_n(x_n)}{\mu_n(x_n)}. \tag{18}$$

An obvious Monte Carlo approximation of μ_n can be obtained based on the current particles

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

but the complexity of the algorithm would then be in $O\left(N^2\right)$ which is prohibitive. Moreover, it might not be possible to compute $\widehat{\mu}_n(x_n)$ pointwise; for instance when K_n is an MH step. The introduction of the artificial distribution $\widetilde{\pi}_n(x_{1:n})$ given by (6) allows us to perform importance sampling without having to compute the marginal distribution (16). The resulting unnormalized importance weights are given by

$$w_n(x_{1:n}) = \frac{\pi_n(x_n)\,\widetilde{\pi}_n(x_{1:n-1}|\,x_n)}{\mu_n(x_{1:n})}. \tag{19}$$

Note that the computation of this weight still generally requires pointwise evaluation of the transition kernels $\{K_n\}$. However, $\{\widetilde{\pi}_n\}$ can be chosen to avoid having to perform these evaluations. Some examples are discussed further.

Avoiding having to compute μ_n comes at the price of extending the integration domain from E to E_n and increasing the variance (if it exists) of the importance sampling estimate. Therefore we are interested in selecting $\{\widetilde{\pi}_n\}$ to minimize the variance of the importance weights on this joint space.

Proposition 2. *The conditional distribution $\widetilde{\pi}_n^{opt}$ on E_{n-1} which minimizes the variance of the importance weight (19) is given by*

$$\widetilde{\pi}_n^{opt}(x_{1:n-1}|\,x_n) = \mu_n(x_{1:n-1}|\,x_n) \tag{20}$$

and this conditional distribution admits the form (7), with for any k,

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

The result of this proposition is actually intuitive and simply states that the optimal distribution (20) takes us back to the case where one performs importance sampling on E instead of E_n as

$$w_n(x_{1:n}) = \frac{\pi_n(x_n)\,\widetilde{\pi}_n^{\mathrm{opt}}(x_{1:n-1}|\,x_n)}{\mu_n(x_{1:n})} = \frac{\pi_n(x_n)}{\mu_n(x_n)}.$$

It is impossible in practice to use $\widetilde{\pi}_n^{\text{opt}}$ as μ_n is unknown. However this suggests that $\widetilde{\pi}_n\left(x_{1:n-1}|\, x_n\right)$ should be chosen to approximate $\widetilde{\pi}_n^{\text{opt}}\left(x_{1:n-1}|\, x_n\right)$. In practice, one cannot typically compute $\{L_n^{\text{opt}}\}$ exactly, but it might be possible to approximate these kernels depending on the context. Some of these approximations will be discussed further. One point used recurrently is that it follows from (19) that

$$w_n\left(x_{1:n}\right) = w_{n-1}\left(x_{1:n-1}\right) \frac{\pi_n\left(x_n\right)}{\pi_{n-1}\left(x_n\right)} \frac{L_{n-1}\left(x_n, x_{n-1}\right)}{K_n\left(x_{n-1}, x_n\right)},$$

so a suboptimal strategy consists of using an L_n which is an approximation of the optimal kernel (21) where one has substituted π_{n-1} for μ_{n-1}; that is we try to approximate

$$L_{n-1}\left(x_n, x_{n-1}\right) = \frac{\pi_{n-1}\left(x_{n-1}\right) K_n\left(x_{n-1}, x_n\right)}{\pi_{n-1} K_n\left(x_n\right)}. \tag{22}$$

This is often more convenient to approximate than (21) as $\{\pi_n\}$ is known up to a normalizing constant whereas $\{\mu_n\}$ is not. Note that if particles have been resampled at time $n-1$, μ_{n-1} is indeed (approximately) equal to π_{n-1} and thus (21) is equal to (22).

Although, it is possible to use other kernels for L_n rather than (21)-(22) or approximations of these expressions, we emphasize that many seemingly attractive choices would lead to inefficient algorithms if one is not careful. We detail three examples below. The connections with algorithms presented in the literature are discussed in the following subsection.

Example 1. By selecting L_{n-1} to equal K_n, the incremental weight (8) looks like a MH ratio. This choice might be "aesthetic" but it is inefficient in most cases. Consider the toy example where $E = \mathbb{R}$, $\pi_1\left(x\right) = \mathcal{N}\left(x; 0, \sigma_1^2\right)$, $\pi_2\left(x\right) = \mathcal{N}\left(x; 0, \sigma_2^2\right)$ and $K_2\left(x, x'\right) = \mathcal{N}\left(x'; x, \sigma_K^2\right)$ with $\sigma_1 \geq \sigma_2$ where $\mathcal{N}\left(z; \mu, \sigma^2\right)$ denotes a Gaussian distribution of argument z, mean μ and variance σ^2. Assume particles at time 1 have been resampled so that $\mu_1 = \pi_1$. Then, whatever σ_K is, the importance weight on the marginal space would be upper bounded

$$\frac{\pi_2\left(x\right)}{\mu_2\left(x\right)} < \infty.$$

and have a finite variance. Now assume that one does not perform importance sampling on E but on the extended space $E \times E$ by introducing an auxiliary kernel $L_1 = K_2$. In this case the importance weight is given by

$$\frac{\pi_2\left(x'\right) K_2\left(x', x\right)}{\pi_1\left(x\right) K_2\left(x, x'\right)} = \frac{\pi_2\left(x'\right)}{\pi_1\left(x\right)}.$$

This is not upper bounded over $E \times E$ and does not admit a finite variance. So if E is not a compact set, one cannot expect this choice to perform well.

Example 2. Consider the case where K_n is an MCMC kernel of invariant distribution π_n. A convenient choice for L_{n-1} is given by

$$L_{n-1}\left(x_n, x_{n-1}\right) = \frac{\pi_n\left(x_{n-1}\right) K_n\left(x_{n-1}, x_n\right)}{\pi_n\left(x_n\right)} \tag{23}$$

for which the incremental weight (8) is independent of x_n and is given by

$$\frac{\pi_n\left(x_{n-1}\right)}{\pi_{n-1}\left(x_{n-1}\right)}. \tag{24}$$

The kernel (23) is a good approximation of (22) if $\pi_{n-1} \approx \pi_n$. In the case where π_n and π_{n-1} differ significantly, one has to be careful when applying this method as it could be very inefficient. Indeed, with L_{n-1} given by (23), the resulting weights are independent of x_n. So even if K_n is fast mixing (in the limiting case $K_n(x, x') = \pi_n(x')$) then the particles are still weighted according to (24). It follows that if resampling is used, it is more efficient to first resample particles with respect to their weights (24) before sampling them according to K_n, instead of using sampling followed by resampling¶. This increases the diversity among particles at time n and it is possible since (24) is independent of x'. This simple approach is attractive but could perform poorly in some applications. Consider a case where the regions of high probability mass for π_{n-1} and π_n are located in disjoint parts of the state space; e.g. a sequential Bayesian estimation problem where π_n is given by (14) and y_n is an informative observation. In this case when the particles are resampled according to (24), only very few will survive and these surviving particles might not even be located in the regions of high probability mass of π_n. Hence, except when K_n is mixing very quickly, performance will not be satisfactory.

Example 3. Consider here the homogeneous case where $\pi_n = \pi$, $K_n = K$, $L_n = L$ where K is an MCMC kernel of invariant distribution π and

$$L(x, x') = \frac{\pi(x') K(x', x)}{\pi(x)}. \tag{25}$$

This is a special case of Example 2 worth detailing. After having resampled the particles once, they are approximately distributed according to π. In this case, (25) is equal to the optimal kernel (21) and the associated importance weights now equal 1; i.e. each particle evolves independently according to K and it is not necessary to make them interact anymore. It may be tempting to consider a kernel L different from (25); in this case the particles would have to be resampled periodically. However the use of such a strategy does not appear justified as resampling is not performed to modify the marginal distribution of the particles but only the correlation between surviving particles at two successive time instants. This limits the diversity in the set of particles and one cannot expect the variance of the resulting estimate (10) to be lower than the one obtained using non-interacting MCMC chains.

Initial distribution μ_1 and sequence of proposal kernels $\{K_n\}_{n \in \mathcal{N} \setminus \{1\}}$.

As in standard importance sampling, the initial distribution μ_1 should be selected such that the importance weight $\pi_1(x)/\mu_1(x)$ is upper bounded over E. In high dimensional problems, if π_1 is not a simple distribution, this might be difficult to ensure. Even if it is satisfied, the ESS criterion might be close to 1. Thus the user has to make sure that it is easy to sample from π_1. Consequently, even if we are interested in sampling from a given target π, we advocate that in complex cases we should use a sequence of distributions moving "slowly" towards π as in (Gelman and Meng, 1998; Neal, 2001).

Let us now give some guidelines on the sequence of proposal kernels to use. It would be tempting to try to re-use the methodology presented in the SMC literature to design efficient sampling schemes K_n but our framework is different. Clearly, the optimal proposal is simply $K_n(x, x') = \pi_n(x')$. As this choice is impossible, we must formulate sensible alternatives.

¶For readers familiar with the SMC literature, this corresponds to the "optimal" importance sampling distribution in (Doucet *et al.*, 2000) and the perfect adaptation method in (Pitt and Shephard, 1999).

Consider the case where $\pi_n = \pi$ for all n. First assume $K_n = K$ is selected as an ergodic MCMC kernel of invariant distribution π. In this very particular case, if one was able to compute the optimal kernels $\{L_n^{\text{opt}}\}$, then the ESS would eventually increase over time and would converge to N as n goes to infinity. However, in practice, either $\{L_n^{\text{opt}}\}$ cannot be computed exactly or another kernel such as (25) is used. In this case, resampling will have to be performed to prevent the degeneracy of the weights. As discussed previously in example 3, after a resampling step which gives particles approximately distributed according to π, it does not appear necessary to make the particles interact anymore. SMC samplers with interacting particles are thus expected to be really useful only when K is *not* an MCMC kernel of invariant distribution π.

Consider now the case where K_n is not an MCMC kernel, then it is sensible to adapt K_n over time using the information provided by the particles at the previous time instants. To simplify notation, we do not make explicit the dependency of K_n on the whole set of particles. In the literature, it has previously been proposed to select $K_n(x, x') = K_n(x')$ as an approximation of π based on previous simulation results; in this case the resulting incremental importance weight is obviously equal to

$$\frac{\pi(x')}{K_n(x')}$$

which corresponds to $L_{n-1}^{\text{opt}}(x', x) = \pi(x)$. An example in which K_n is dependent on all particles at the previous time step is given in (West, 1993). West (1993) approximates the distribution of the particles by a mixture of t-distributions which is then used as a new importance function. These algorithms may fail when the target is multimodal with well-separated narrow modes. Indeed in this case the probability of obtaining samples in the relevant zones of the space is very small and an importance distribution based on these particles could be inefficient. Therefore, for difficult scenarios, it is unlikely that such approaches will be robust and the introduction of an intermediate sequence of distributions $\{\pi_n\}$ to move smoothly to π is again recommended.

Finally, consider the case where π_n varies over time. In this framework, the use at time n of an MCMC kernel of invariant distribution π_n is justified either if K_n is fast mixing and/or π_n is varying slowly over time so that one can expect the current distribution of particles to be reasonably close to the target distribution. In cases where π_n is evolving quickly over time and K_n mixes slowly, the use of an MCMC kernel is not necessarily appropriate and it might be preferable to use a random walk proposal biased towards the regions of high probability masses of the target distribution of interest. In the simulation part, we discuss a sequential Bayesian inference problem where it is useful to use a kernel K_n which is not an MCMC kernel and where it is possible to approximate (22).

2.4. Connections to previous work and Extensions

Connections to previous work. AIS is a method proposed recently by Neal (2001). An essentially similar idea was developed independently in physics (Jarzinsky, 1997). Reversing the time index in (Neal, 2001) to be consistent with our notation, AIS corresponds to the case where one considers a finite sequence of distributions $\{\pi_n\}$ given by (13), K_n is an MCMC kernel of invariant distribution π_n and L_{n-1} is given by (23). No resampling step is used in AIS. With this specific choice for $\{\pi_n\}$, the variance of the importance weights can indeed converge to a finite limit as the number p of intermediate distributions

goes to infinity even if resampling is not used. Loosely speaking, this is the case because π_n is getting very close to π_{n-1} and π_p is not a degenerate distribution. It is expected however that resampling will improve the variance of the Monte Carlo estimates and this is demonstrated in the simulation section. This algorithm with resampling has also been used in the context of optimal filtering by (Godsill and Clapp, 2001).

Chopin (2002) considers the sequential Bayesian inference case where $\{\pi_n\}$ is given by (14). His algorithm corresponds to the case where K_n is an MCMC kernel of invariant distribution π_n and L_{n-1} is given by (23); i.e. it is very similar to AIS except that a resampling step is included. As discussed earlier, in this case resampling is very useful as π_{n-1} and π_n can differ significantly; see also (Gilks and Berzuini, 2001) for a closely related work in the filtering context.

A more recent work (Cappé *et al.*, 2004) is another special case of this framework. The authors consider the homogeneous case where $\pi_n = \pi$, $K_n = K$ and $L_n = L$. Their algorithm corresponds to the case where K is an MCMC kernel of invariant distribution π (namely a Gibbs sampler) and $L(x, x') = \pi(x')$. They also propose the use of a kernel $K_n(x, x') = K_n(x')$ where the parameters of the importance distribution K_n are determined using statistics over the whole population of particles at time $n-1$.

Discussion and Extensions. The algorithm described in this section must be interpreted as the basic element of more complex algorithms. It is what the MH algorithm is to MCMC. For complex MCMC problems, one typically uses a combination of MH steps where the n_x components of x say $(x_1, \ldots, x_{n_x})$ are updated by subblocks (Robert and Casella, 1999). Similarly, to sample from high dimensional distributions, a practical SMC sampler can update the components of x via subblocks. A mixture of transition kernels can also be used at each time n. Let us assume $K_n(x, x')$ is of the form

$$K_n(x, x') = \sum_{m=1}^{M} \alpha_{n,m}(x) K_{n,m}(x, x') \tag{26}$$

where $\alpha_{n,m}(x) > 0$, $\sum_{m=1}^{M} \alpha_{n,m}(x) = 1$ and $\{K_{n,m}\}$ is a collection of transition kernels. In this case, the incremental weights can be computed by the standard formula (3). However, this could be too expensive if M is large. An alternative valid approach consists of using

$$\frac{\pi_n\left(X_n^{(i)}\right) \beta_{n-1,M_n^{(i)}}\left(X_n^{(i)}\right) L_{n-1,M_n^{(i)}}\left(X_n^{(i)}, X_{n-1}^{(i)}\right)}{\pi_{n-1}\left(X_{n-1}^{(i)}\right) \alpha_{n,M_n^{(i)}}\left(X_{n-1}^{(i)}\right) K_{n,M_n^{(i)}}\left(X_{n-1}^{(i)}, X_n^{(i)}\right)} \tag{27}$$

where $M_n^{(i)}$ is the discrete random variable such that $\Pr\left(M_n^{(i)} = m\right) = \alpha_{n,m}\left(X_{n-1}^{(i)}\right)$ and $X_n^{(i)} \sim K_{n,M_n^{(i)}}\left(X_{n-1}^{(i)}, \cdot\right)$. In (27), $\beta_{n-1,m}(x) > 0$, $\sum_{m=1}^{M} \beta_{n-1,m}(x) = 1$ and $\{L_{n-1,m}\}$ is a collection of transition kernels. It is straightforward to determine the optimal choice for $\{\beta_{n-1,m}\}$ and $\{L_{n-1,m}\}$ in terms of minimization of the variance of weights. By following arguments similar to Subsection 2.3, one obtains

$$\beta_{n-1,m}^{\mathrm{opt}}(x') L_{n-1,m}^{\mathrm{opt}}(x', x) = \frac{\pi_{n-1}(x) \alpha_{n,m}(x) K_{n,m}(x, x')}{\sum_{m=1}^{M} \int \pi_{n-1}(x) \alpha_{n,m}(x) K_{n,m}(x, x') dx}. \tag{28}$$

The mixture of transition kernels is very useful in cases where we are interested in sampling distributions defined on an union of subspaces of different dimensions, where say $E =$

$\cup_{k=0}^{\infty} \{k\} \times \vartheta_k$; i.e. in a context where Reversible Jump MCMC (RJMCMC) algorithms are traditionally used (Green, 1995). In these cases, we obtain trans-dimensional SMC algorithms. In this context, the scenario where $\{\pi_n\}$ has support D_n with $D_{n-1} \subseteq D_n \subseteq E$ for any n, is of special interest. An example is discussed in the simulation section.

There are also numerous potential extensions. As suggested in (Crisan and Doucet, 2000), (Cappé *et al.*, 2004), (Chopin, 2002) or (West, 1993), one can use a proposal kernel whose parameters are a function of the whole set of current particles. This allows the algorithm to automatically scale the proposal distribution based on the previous importance weights.

3. Simulation Results

We demonstrate the performance of SMC samplers on two problems arising in Bayesian statistics; namely Bayesian variable selection and sequential Bayesian estimation of the intensity of an inhomogeneous Poisson process.

3.1. Bayesian Variable Selection

For any $(X, Y) \in \mathcal{X} \times \mathbb{R}$, we consider the following regression model (Kohn *et al.*, 2001)

$$Y = \sum_{k=1}^{M} I_k \beta_k \Psi_k(X) + V;\ V \sim \mathcal{N}\left(0, \sigma^2\right).$$

The indicator variable $I_k \in \{0, 1\}$ is such that $\beta_k = 0$ if $I_k = 0$ and $\beta_k \neq 0$ if $I_k = 1$; i.e. there are 2^M different models for the regression function. Assuming T independent identically distributed data points $(X_{1:T}, Y_{1:T})$ are available, one has in a vector-matrix form

$$Y_{1:T} = D(I_{1:M})\, \beta(I_{1:M}) + V_{1:T},$$

where $D(I_{1:M})$ is a $T \times l(I_{1:M})$ matrix and $l(I_{1:M}) = \sum_{k=1}^{M} I_k$ is the number of basis terms included in the model. The j^{th} column of $D(I_{1:M})$ corresponds to $(\Psi_{\alpha(I_{1:M},j)}(X_1), \ldots, \Psi_{\alpha(I_{1:M},j)}(X_T))^{\mathrm{T}}$ where $\alpha(I_{1:M}, j)$ is the index of the j^{th} non-null coefficient of the sequence $I_{1:M}$ and $\beta_{I_{1:M}}$ is the associated $l(I_{1:M})$-dimensional vector of non-null regression coefficients. To complete our model, we set

$$\begin{aligned} \beta(I_{1:M}) | \left(\sigma^2, I_{1:M}\right) &\sim \mathcal{N}\left(0, \delta^2 \sigma^2 \left(D^{\mathrm{T}}(I_{1:M}) D(I_{1:M})\right)^{-1}\right), \\ \sigma^2 &\sim \mathcal{IG}\left(\frac{\gamma_0}{2}, \frac{\nu_0}{2}\right), \end{aligned}$$

and $\Pr(I_k = 1 | \lambda) = \lambda$ where λ is uniformly distributed on $[0, 1]$. Finally γ_0, ν_0 and δ are fixed hyperparameters. Given a realization $(x_{1:T}, y_{1:T})$, we are interested in both sampling and maximizing the marginal posterior distribution

$$p(i_{1:M} | x_{1:T}, y_{1:T}) \propto \left(\nu_0 + y_{1:T}^{\mathrm{T}} P(i_{1:M})\, y_{1:T}\right)^{T/2 + \frac{\gamma_0}{2}} \left(1 + \delta^2\right)^{-l(i_{1:M})/2} l(i_{1:M})!\,(T - l(i_{1:M}))!$$

where

$$P(i_{1:M}) = I_{l(i_{1:M})} - \left(1 + \delta^{-2}\right)^{-1} D(i_{1:M}) \left(D^{\mathrm{T}}(i_{1:M}) D(i_{1:M})\right)^{-1} D^{\mathrm{T}}(i_{1:M})$$

with $I_{l(i_{1:M})}$ the identity matrix of dimension $l(i_{1:M})$. The data are taken to be the sinc function, i.e. $\mathrm{sinc}(x) = \sin(x)/x$, corrupted by additive Gaussian noise with $\sigma = 0.1$ for $T = 50$ evenly spaced points in the interval $[-10, 10]$. We select $M = T$ basis functions of the form

$$\Psi_k(x) = \frac{1}{\sqrt{2\pi}\phi} \exp\left(-\frac{(x - x_k)^2}{2\phi^2}\right)$$

where $\phi = 1.6$.

To sample from $p(i_{1:M} | y_{1:T}, x_{1:T})$, we consider the sequence of distributions

$$\pi_n(i_{1:M}) \propto [p(i_{1:M} | x_{1:T}, y_{1:T})]^{\gamma_n} \tag{29}$$

where $n \in \{1, \ldots, p\}$ and $p \in \{250, 500, 1250, 2500, 5000\}$. For the schedule, we set $\gamma_1 = 0$ (i.e. $\pi_1 = \mu_1$ is the uniform distribution). The sequence $\{\gamma_n\}$ initially increases linearly for $\lfloor \frac{p}{5} \rfloor$ steps and then according to $a \log(n) + b$ with $\gamma_p = 1$. To carry out optimization and explore the modes of $p(i_{1:M} | x_{1:T}, y_{1:T})$, we consider also a sequence of distributions of the form (29) where $n \in \{1, \ldots, p\}$ and $p \in \{250, 500, 1250, 2500, 5000\}$. In this case, we select have a small linear schedule for $\lfloor \frac{p}{3} \rfloor$ initially and then use $a \log(n) + b$ with $\gamma_1 = 0$ and $\gamma_p = 10$. In all cases, we select K_n as a deterministic scan Gibbs sampler of invariant distribution π_n. Only one variable is updated at each iteration, hence $p \gg 1$. For the kernel L_n, we consider both (22) and the AIS choice (23). For a particle $I_{1:M}^{(n-1)}$ (resp. $I_{1:M}^{(n)}$) at time $n-1$ (resp. n), the incremental weights for AIS and (22) are respectively given by

$$\frac{\pi_n\left(I_{1:M}^{(n-1)}\right)}{\pi_{n-1}\left(I_{1:M}^{(n-1)}\right)} \text{ and } \frac{\pi_n\left(I_{1:M}^{(n-1)}\right) + \pi_n\left(I_{1:M}^{(n)}\right)}{\pi_{n-1}\left(I_{1:M}^{(n-1)}\right) + \pi_{n-1}\left(I_{1:M}^{(n)}\right)}.$$

In this case, and more generally in any discrete state-space problems with local exploration, it is usually possible to compute (22) exactly‖. Note that given the expression of the weights, one cannot expect much improvement from (22) over AIS when $\pi_n \approx \pi_{n-1}$. The computational complexity of AIS and of the alternative method we propose is similar.

We test both algorithms with resampling and without resampling using $N = 1000$ particles. Resampling is performed when the ESS is below $N/2$. We compare our algorithm to sampling from π with a Gibbs sampler, using pN iterations for the computational complexity of both methods to be approximately similar. To optimize π, we compare the SMC samplers with simulated annealing versions of the Gibbs sampler. Two scenarios are considered: first we consider a long annealing run such that $\gamma_1 = 0$ and $\gamma_{pN} = 10$, second we consider N parallel (non-interacting) annealing runs such that $\gamma_1 = 0$ and $\gamma_p = 10$.

In Table 1, we display the average Mean Square Error (MSE) and the standard deviation of the MSE estimate of the regression function over 50 simulations using the same dataset. We also display the average and standard deviation of the mean of the log-posterior of the last population of particles. For MCMC we give the average and standard deviation of the mean of the log-posterior of the samples obtained after burn-in. The results are presented for AIS using (22) or (23), for SMC using (23) and finally for the Gibbs sampler. For the MCMC results, we discard the first 40% of samples as burn-in period. For each simulation, the same N random initial starting points are used for AIS and SMC and one of those

‖It would be possible to compute analytically $\pi_{n-M} K_{n-M+1:n}(i_{1:K})$ to reduce further the variance but the computational complexity increases exponentially with M.

N points is used to initialize the Gibbs sampler. The results are presented for AIS using (22) or (23), for SMC using (23) and finally for the Gibbs sampler. As expected, there is almost no difference between AIS using (22) or (23). The results demonstrate that in all simulations, the resampling step used in the SMC algorithm produces a reduction in the variance cheaply. The reduction of the variance is most prominent when the number of updates per site is small, hence p is small. Intuitively this makes sense since in these situations the difference between π_{n-1} and π_n can be significant when compared to the situations in which p is very large and $\pi_{n-1} \approx \pi_n$. An additional point to mention is that, as we would expect, the number of times resampling is carried out increases as p decreases. Finally, for large p where $\pi_{n-1} \approx \pi_n$, the SMC algorithm and AIS give almost similar results with regard to the average MSE and its standard deviation. However the average log-posterior for the final population of samples is clearly higher for SMC compared to AIS. Compared to SMC, MCMC algorithms yield a lower average MSE. Nevertheless, we just use one realization of observations so this is not significant. For example, the average MSE appeared to be unchanged at approximately around 2.2 even for low values of the average mean log posterior. For a small p, SMC yields samples with higher log-posterior values. As p increases, the average mean log-posterior is higher for MCMC than for SMC but the variance for MCMC remains significantly higher in all cases.

MCMC					
avg. MSE	2.29	2.32	2.49	2.48	2.21
std. MSE	0.93	0.83	0.91	0.81	0.71
avg. mean log posterior	-0.33	3.57	5.81	6.31	6.26
std. mean log posterior	3.61	4.54	4.75	2.92	2.22
	Updates per site				
	5	10	25	50	100
AIS with (23)					
avg. MSE	4.79	3.26	3.50	3.29	3.44
std. MSE	2.83	1.25	1.60	1.08	1.06
avg.mean log posterior last population	-7.17	-3.21	-0.76	0.90	2.12
std. mean log posterior last population	0.24	0.17	0.23	0.20	0.26
avg. ESS last population	3.50	4.50	13.35	79.21	85.21
AIS with (22)					
avg. MSE	4.78	3.26	3.50	3.28	3.44
std. MSE	2.83	1.25	1.60	1.08	1.06
avg. ESS last population	3.52	4.51	13.39	81.27	87.89
SMC with (23)					
avg. MSE	3.05	3.04	3.34	3.19	3.65
std. MSE	1.45	1.22	1.17	0.93	1.01
avg. mean log posterior last population	2.53	3.84	4.09	4.63	4.51
std. mean log posterior last population	2.21	0.67	1.56	0.42	0.69
avg. number of resampling steps	7.86	7.82	6.46	4.98	3.22
avg. ESS last population	820.97	925.10	756.96	880.23	802.22

Table 1: Performance of MCMC, AIS and SMC over 50 simulations

In Table 2, we display the average log-posterior values of the estimated mode and its standard deviations over 50 simulations. Again we use the same data set and the same

initialization procedure. The posterior mode estimate used to obtain the results for each algorithm was chosen as the sample generated during the simulation which maximized the posterior distribution. We compare SMC to a long run of annealing and to multiple non-interacting parallel annealing runs. The SMC algorithm outperforms both these techniques and again this is especially evident when p is small. This is best demonstrated by the simulations where there are 5 updates per site, which corresponds to $p = 250$. In all our simulations, the best estimated mode had a log posterior value of 14.31. In this case the number of times the SMC algorithm reaches a maximum log-posterior value equal to 14.31 is more than twice as often as parallel annealing and the long run annealing never obtains a maximum log-posterior value even close to it. This further emphasises the gains that can be made through the introduction of resampling which allows the simulated annealing chains to interact in a principled manner.

Long run Annealing					
avg. max log posterior mode	5.07	7.59	8.90	11.12	12.13
std. max log posterior mode	3.24	3.21	3.22	2.74	2.40
number of times reached mode14.31	0	1	8	17	24
	Updates per site				
	5	10	25	50	100
SMC optimization with (23)					
avg. max log posterior mode	11.67	13.15	14.26	14.31	14.31
std. max log posterior mode	2.35	1.47	0.34	0.00	0.00
number of times reached mode 14.31	14	30	49	50	50
Parallel Annealing runs					
avg. max log posterior mode	11.51	12.40	14.15	14.26	14.31
std. max log posterior mode	1.41	1.43	0.66	0.34	0.00
number of times reached mode 14.31	6	16	46	49	50

Table 2: Performance of simulated annealing and SMC over 50 simulations

3.2. Sequential Bayesian Estimation for an Inhomogeneous Poisson Process

Let us consider the following model. At time t, we have access to time occurrences which are assumed to follow an inhomogeneous Poisson process of intensity $\lambda : \mathbb{R}^+ \to \mathbb{R}^+$; that is at time t the likelihood of l_t time occurrences is given by

$$p_t\left(y_{1:l_t} \middle| \{\lambda(u)\}_{u \leq t}\right) = \exp\left(-\int_0^t \lambda(u)\, du\right) \prod_{l=1}^{l_t} \lambda(y_l).$$

We are interested in estimating the unknown intensity $\lambda(t)$ sequentially in time. We adopt a simple piecewise constant model for $\lambda(t)$

$$\lambda(t) = \sum_{m=1}^{k} \lambda_m \mathbb{I}_{[\tau_{m-1}, \tau_m)}(t) + \lambda_{k+1} \mathbb{I}_{[\tau_m, \infty)}(t)$$

with $\tau_0 = 0$. The number of steps k, their amplitudes $\lambda_{1:k+1}$ and the knot points $\tau_{1:k}$ are assumed unknown. We use the following time-dependent prior distribution on these unknown parameters

$$p_t(k, \lambda_{1:k+1}, \tau_{1:k}) = p_t(k)\, p_t(\lambda_{1:k+1} | k)\, p_t(\tau_{1:k} | k)$$

where $p_t(k)$ is a Poisson distribution of parameter $\lambda_q t$, $p_t(\tau_{1:k}|\, k)$ is the vector of uniform order statistics on $[0,t)$ and

$$p_t(\lambda_{1:k}|\, k) = p(\lambda_1) \prod_{l=2}^{k+1} p(\lambda_l|\, \lambda_{l-1})$$

where $\lambda_1 \sim \mathcal{G}(\mu, \upsilon)$ and $\lambda_l|\, \lambda_{l-1} \sim \mathcal{G}\left(\lambda_{l-1}^2/\chi; \lambda_{l-1}/\chi\right)$; μ, υ, χ are parameters specified by the user.

We are interested here in estimating the sequence of posterior distributions over times $n\Delta T$

$$\begin{aligned} \pi_n(k, \lambda_{1:k+1}, \tau_{1:k}) &= p_{n\Delta T}(k, \lambda_{1:k+1}, \tau_{1:k}|\, y_{1:l_{n\Delta T}}) \\ &\propto p_{n\Delta T}(y_{1:l_{n\Delta T}}|\, k, \lambda_{1:k+1}, \tau_{1:k})\, p_{n\Delta T}(k, \lambda_{1:k+1}, \tau_{1:k}) \end{aligned}$$

where ΔT is a time interval defined by the user. These distributions are defined on $E = \cup_{k=0}^{\infty} \{k\} \times \vartheta_k$ where $\vartheta_k = \left\{\tau_{1:k} \in \mathbb{R}^k; 0 < \tau_1 < \ldots < \tau_k\right\} \times (\mathbb{R}^+)^{k+1}$, the support of π_n being reduced to the subset $\left\{\tau_{1:k} \in \mathbb{R}^k; 0 < \tau_1 < \ldots < \tau_k < n\Delta T\right\} \times (\mathbb{R}^+)^{k+1}$.

This is a problem where the number of unknowns is something you do not know. To sample from one of these distributions, a standard approach would consist of using a Reversible Jump MCMC algorithm (Green, 1995). We propose here to sample instead from the sequence of distributions using SMC samplers. At each time step, we consider using a mixture of four different moves.

The first move consists of not doing anything; i.e. $K_{n,1}\left((k, \lambda_{1:k+1}, \tau_{1:k}), \left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)\right) = \delta_{k,\lambda_{1:k+1},\tau_{1:k}}\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)$ so the incremental weight is given by

$$\frac{\pi_n\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)}{\pi_{n-1}\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)}. \tag{30}$$

Given $(k, \lambda_{1:k+1}, \tau_{1:k})$, the second move is a birth move

$$K_{n,2}\left((k, \lambda_{1:k+1}, \tau_{1:k}), \left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)\right) = \delta_{k+1,\lambda_{1:k+1},\tau_{1:k}}\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right) q_n\left((\lambda_{1:k+1}, \tau_{1:k}), \left(\lambda'_{k+2}, \tau'_{k+1}\right)\right)$$

where the appended component $\left(\lambda'_{k+2}, \tau'_{k+1}\right)$ is sampled according to a proposal distribution $q_n((\lambda_{1:k}, \tau_{1:k}), \cdot)$. In this case, it is possible to compute (22) and the incremental weight is then given by

$$\frac{\pi_n\left(k', \lambda'_{1:k'+2}, \tau'_{1:k'+1}\right)}{\pi_{n-1}(k, \lambda_{1:k+1}, \tau_{1:k})\, q_n\left((\lambda_{1:k+1}, \tau_{1:k}), \lambda'_{k+2}, \tau'_{k+1}\right)}. \tag{31}$$

It is easy to establish that the proposal q_n^{opt} minimizing the variance of this incremental weight, given $(\lambda_{1:k+1}, \tau_{1:k})$, is given by

$$\begin{aligned} &\pi_n\left(\lambda'_{k+2}, \tau'_{k+1}\middle|\, k+1, \lambda_{1:k+1}, \tau_{1:k}\right) \\ = \;& p_{n\Delta T}\left(\tau'_{k+1}\middle|\, y_{1:l_{n\Delta T}}, k+1, \lambda_{1:k+1}, \tau_{1:k}\right) p_{n\Delta T}\left(\lambda'_{k+2}\middle|\, y_{1:l_{n\Delta T}}, k+1, \lambda_{1:k+1}, \tau_{1:k+1}\right) \end{aligned}$$

It is difficult to sample from this optimal distribution and impossible to compute the associated importance weight. Therefore, we propose an approximation of this distribution. First τ'_{k+1} is sampled from a truncated exponential distribution on $[\tau_k, n\Delta T)$. Second, to sample λ_{k+2} using information from the observations, we substitute for the

true likelihood the likelihood which would be obtained if we use the number of counts in intervals of length ΔT as opposed to occurrence times. We can then approximate $p_{n\Delta T}\left(\lambda_{k+2}|\, y_{1:l_{n\Delta T}}, k+1, \lambda_{1:k+1}, \tau_{1:k+1}\right)$ by a Gamma distribution.

Given $(k+1, \lambda_{1:k+2}, \tau_{1:k+1})$, the third move is a death move where we propose to remove a knot (λ_{J+1}, τ_J) among the D most recent knots to obtain $\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right) = \left(k, \lambda_{1:k+2} \backslash \{\lambda_{J+1}\}, \tau_{1:k+1} \backslash \{\tau_J\}\right)$. We sample $J \sim \mathcal{U}\left(\{k-D+1, \ldots . k\}\right)$ according to a uniform distribution. In this case, we can compute (22) and the resulting incremental importance weight is given by

$$\frac{\pi_n\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)}{\pi_{n-1}\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right) D^{-1}}. \tag{32}$$

Finally the fourth and last move is a height adjustment move where we modify the current value of an intensity parameter in the recent past to obtain $\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)$. We sample $J \sim \mathcal{U}\left(\{k-D+1, \ldots . k\}\right)$ then the new intensity λ'_{J+1} according to a discrete probability distribution with support $\{\lambda_{J+1} - s\delta, \lambda_{J+1} - (s-1)\delta, ..., \lambda_{J+1} + s\delta\}$, where s and δ are specified by the user. The discrete proposal distribution** for the intensity to be adjusted is given by

$$q_n\left(\lambda_{J+1}, \lambda'_{J+1}\right) \propto \pi_n\left(k, \lambda_{1:k+1} \backslash \{\lambda_{J+1}\}, \lambda'_{J+1}, \tau_{1:k}\right)$$

and the resulting incremental importance weight using (22) is given by

$$\frac{\pi_n\left(k, \lambda_{1:k+1} \backslash \{\lambda_{J+1}\}, \lambda'_{J+1}, \tau_{1:k}\right)}{\sum_i q\left(\lambda_{J+1}, \lambda_{J+1} - (s-i+1)\delta\right) \pi_{n-1}\left(k, \lambda_{1:k+1} \backslash \{\lambda_{J+1}\}, \lambda_{J+1} - (s-i+1)\delta, \tau_{1:k}\right)}. \tag{33}$$

The expressions (30) to (33) for the importance weights do not include the moves probabilities. The simulations were run using a birth probability $\alpha_{n,2}(k) = C \min\left(1, \frac{\lambda_q t}{k+1}\right)$, a death probability $\alpha_{n,3}(k) = C \min\left(1, \frac{k}{\lambda_q t}\right)$, a height adjustement probability $\alpha_{n,4} = 0.15$ and finally the probability of not moving was selected such that the probabilities sum to 1. The constant C was selected as large as possible under the constraint that $\alpha_{n,2}(k) + \alpha_{n,3}(k) \leqslant 0.85$. These probabilities correspond to the terms $\alpha_{n,m}$ appearing in (26). Using the expression (27) with $\beta_{n,m} = \alpha_{n,m}$ in this example provided computational savings and satisfactory results so we did not approximate the optimal $\beta_{n,m}$ given in (28).

We apply this algorithm to the popular coal mining data set representing coal mine disasters between 1851 and 1962. The version of the data set found in (Green, 1995) was used. However, the analysis carried out was for disasters renormalized on a year scale. We also implement a RJMCMC algorithm based on exactly the same statistical model to sample from the posterior distribution given the whole dataset. The moves utilised in the RJMCMC algorithm were designed to be similar to those used in (Green, 1995). The user specified parameters used were $\Delta T = 1$ year, $\lambda_q = 1/4$, $\mu = 9/2$, $\upsilon = 3/2$, $D = 4$, $s = 4$, $\delta = 0.2$ and $\chi = 0.1$. The number of particles used for the SMC simulation was $N = 25000$ while the RJMCMC algorithm used 220000 samples with the first 20000 samples discarded for the "burn in" stage.

We display in Figure 1 the smoothed estimate of the inhomogeneous Poisson intensity obtained using the SMC algorithm versus the estimated using RJMCMC. The smoothed

**This proposal distribution can be interpreted as a random walk with discrete increments but the marginal distribution of the particles still has a support covering $\mathbb{R}^+$.

SMC estimate presented is $E\left[\lambda\left(t\right)|y_{1:l_n\Delta T+14\Delta T}\right]$, hence it is different from RJMCMC which uses the whole data set. For both the SMC algorithm and the RJMCMC algorithm the estimated $\pm 3\sigma$ error lines are plotted. We also implemented an algorithm based on the methodology presented in Chopin (2002) adapted to the trans-dimensional case; i.e. we use RJMCMC moves for K_n and used (23). The results were not encouraging. We believe the reason for this was that the discrepancy between two successive distributions can be high in this case. Consequently, the effective sample size often dropped to very low values. Our approach allows the user to modify the locations of particles in light of the new observations before reweighting them. In our simulations the ESS never went below $0.3N$.

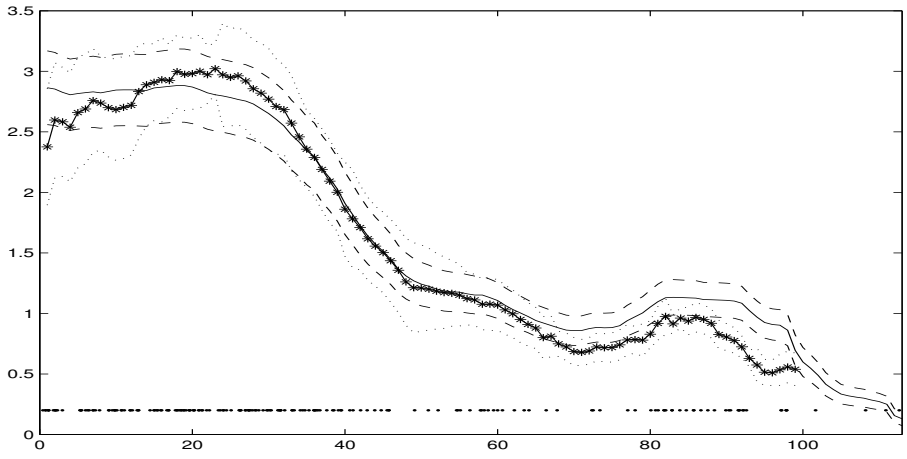

Fig. 1. Bottom: Coal mining disaster data, 1851-1962: occurrences of disasters, Solid line: RJMCMC estimate of the intensity, Dashed lines: RJMCMC estimate ±3 standard deviation, Star: SMC estimate of the intensity, Dotted lines: SMC estimate ±3 standard deviation.

4. Conclusion

In this article, we have presented a class of methods to sample from distributions known up to a normalizing constant and defined on a common space. These methods are based upon SMC algorithms. This framework is flexible and very general. In particular, it yields simple strategies to make parallel MCMC runs interact, and also new algorithms to perform global optimization or sequential Bayesian inference. Simulations demonstrate that this set of methods is potentially powerful. However, there are still several important open methodological and theoretical problems to study.

From a methodological point of view, it would be important to develop efficient generic methods to approximate the optimal auxiliary kernels $\{L_n\}$ especially for latent variable models. In the spirit of path sampling (Gelman and Meng, 1998), it would also be interesting to obtain the optimal path†† for going from an easy to sample distribution π_1 to a fixed

††Optimal in the sense it minimizes the variance of the importance weights.

target distribution $\pi_p = \pi$. Finally, for a fixed computational complexity, there is a trade-off between the number of particles N and the length p of runs. Clearly if the kernels $\{K_n\}$ are mixing well, we should favour shorter runs with many particles whereas if they mix slowly we should have longer runs with less particles. However, it would be of interest to devise quantitative measures. Finally, from a theoretical point of view, it would be of interest to weaken the assumptions in (Del Moral and Doucet, 2003).

5. Acknowledgments

The authors are grateful to their colleagues for their comments. They also would like to acknowledge the reviewers for their comments which allow 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.

6. Appendix

Proof of Proposition 1. The expression (11) follows from the delta method. Expression (12) follows from a convenient rewriting of the variance expression established in (Del Moral, 2004; section 9.4, pp. 300-306); see also (Chopin, 2004; theorem 1) for an alternative derivation. We adopt here the notation of Chopin (2004). The variance is given by

$$\sigma^2_{SMC,n}(\varphi) = E_{\mu_1}\left[w_1^2 \mathcal{E}^2_{2:n}(\varphi - E_{\pi_n}(\varphi))\right] + \sum_{k=2}^{n} E_{\pi_{k-1}K_k}\left[w_k^2 \mathcal{E}^2_{k+1:n}(\varphi - E_{\pi_n}(\varphi))\right] \quad (34)$$

where $\mathcal{E}_{n+1:n}(\varphi) = \varphi$,

$$\mathcal{E}_{k+1:n}(\varphi) = \mathcal{E}_{k+1} \circ \cdots \circ \mathcal{E}_n(\varphi)$$

and

$$\mathcal{E}_n(\varphi)(x_{n-1}) = E_{K_n(x_{n-1},\cdot)}\left[w_n(x_{n-1}, X_n)\varphi(X_n)\right]$$

where $w_n(\cdot,\cdot)$ is defined in (8). The expression (34) is difficult to interpret. It is conveniently rearranged here. The key is to notice that

$$\begin{aligned}
\mathcal{E}_n(\varphi)(x_{n-1}) &= E_{K_n(x_{n-1},\cdot)}\left[w_n(x_{n-1}, X_n)\varphi(X_n)\right] \\
&= \int K_n(x_{n-1}, x_n) \frac{\pi_n(x_n) L_{n-1}(x_n, x_{n-1})}{\pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)} \varphi(x_n)\, dx_n \\
&= \frac{1}{\pi_{n-1}(x_{n-1})} \int \varphi(x_n)\, \pi_n(x_n)\, L_{n-1}(x_n, x_{n-1})\, dx_n. \\
&= \frac{\widetilde{\pi}_n(x_{n-1})}{\pi_{n-1}(x_{n-1})} \int \varphi(x_n)\, \widetilde{\pi}_n(x_n | x_{n-1})\, dx_n
\end{aligned}$$

Similarly, one obtains

$$\begin{aligned}
& \mathcal{E}_{n-1:n}(\varphi) \\
= \; & \mathcal{E}_{n-1}(\mathcal{E}_n(\varphi))(x_{n-2}) \\
= \; & E_{K_{n-1}(x_{n-2},\cdot)}[w_{n-1}(x_{n-2:n-1})\,\mathcal{E}_n(\varphi)(x_{n-1})] \\
= \; & \frac{1}{\pi_{n-2}(x_{n-2})}\int\left(\frac{1}{\pi_{n-1}(x_{n-1})}\int\varphi(x_n)\,\pi_n(x_n)\,L_{n-1}(x_n,x_{n-1})\,dx_n\right) \\
& \qquad \times \pi_{n-1}(x_{n-1})\,L_{n-2}(x_{n-1},x_{n-2})\,dx_{n-1}. \\
= \; & \frac{1}{\pi_{n-2}(x_{n-2})}\int\left(\int\varphi(x_n)\,\widetilde{\pi}_n(x_{n-1:n}|\,x_{n-2})\,dx_{n-1:n}\right)\widetilde{\pi}_{n-2}(x_{n-2})\,dx_{n-1}. \\
= \; & \frac{\widetilde{\pi}_{n-1}(x_{n-2})}{\pi_{n-2}(x_{n-2})}\int\varphi(x_n)\,\widetilde{\pi}_n(x_n|\,x_{n-2})\,dx_n
\end{aligned}$$

and, by induction, one gets

$$\begin{aligned}
\mathcal{E}_{k+1:n}(\varphi) &= \frac{1}{\pi_k(x_k)}\varphi(x_n)\,\pi_n(x_n)\prod_{i=k}^{n-1}L_i(x_i,x_{i-1})\,dx_{k+1:n}. \qquad (35)\\
&= \frac{\widetilde{\pi}_n(x_k)}{\pi_k(x_k)}\int\varphi(x_n)\,\widetilde{\pi}_n(x_n|\,x_k)\,dx_n.
\end{aligned}$$

The expression of $\sigma^2_{SMC,n}(\varphi)$ given (12) follows now directly from (35) and (34).

Proof of Proposition 2. The result follows easily from the variance decomposition formula

$$var[w(X_{1:n})] = E[var[w(X_{1:n})|\,X_n]] + var[E[w(X_{1:n})|\,X_n]]. \qquad (36)$$

The second term on the right hand side of (36) is independent of $\widetilde{\pi}_n(x_{1:n-1}|\,x_n)$ as

$$E[w(X_{1:n})|\,X_n] = \frac{\pi_n(X_n)}{\mu_n(X_n)}$$

whereas $var[w(X_{1:n})|\,X_n]$ is equal to zero if using (20). It is straightforward to check that (20) admits the form (7) for $\{L_n\}$ given by (21), i.e.

$$\mu_1(x_1)\prod_{k=2}^{n}K_k(x_{k-1},x_k) = \mu_n(x_n)\prod_{k=2}^{n}\frac{\mu_{k-1}(x_{k-1})\,K_k(x_{k-1},x_k)}{\mu_k(x_k)}. \qquad (37)$$

Note that (37) is simply the forward-backward formula for Markov processes.

References

[1] Cappé, O. Guillin, A., Marin, J.M. and Robert, C.P. (2004) Population Monte Carlo. *J. Comp. Graph. Stat.*, to appear.

[2] Chopin, N., (2002) A sequential particle filter method for static models, *Biometrika*, **89**, 539-552.

[3] Chopin, N., (2004) Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference, *Annals of Statistics*, to appear.

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

[5] Crisan, D. (2001) Particle filters - A theoretical perspective. In *Sequential Monte Carlo Methods in Practice,* Ed. Doucet, A., De Freitas, J.F.G and Gordon, 17-38.

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

[7] Del Moral, P. and Doucet, A. (2003) On a class of genealogical and interacting Metropolis models. In *Séminaire de Probabilités XXXVII*, Ed. Azéma, J., Emery, M., Ledoux, M. and Yor, M., *Lecture Notes in Mathematics*, Berlin: Springer-Verlag, **1832,** 415-446.

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

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

[10] Gelman, A. and Meng, X.L. (1998) Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Stat. Science*, **13**, 163-185.

[11] Geyer, C.J. and Thompson, E.A. (1995) Annealing Markov chain Monte Carlo with applications to ancestral inference. *J. Am. Statist. Ass.*, **90**, 909-920.

[12] 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.

[13] Godsill, S.J. and Clapp, T. (2001) Improvement strategies for Monte Carlo particle filters. In *Sequential Monte Carlo Methods in Practice,* Ed. Doucet, A., De Freitas, J.F.G and Gordon, N.J., 139-158.

[14] Goldberg D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, Massachusetts.

[15] Green, P.J. (1995) Reversible jump MCMC computation and Bayesian model determination. *Biometrika*, **82**, 711-732.

[16] Jarzynski, C. (1997) Nonequilibrium equality for free energy differences. *Physical Review Letters*, **78**, 2690-2693.

[17] Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, **5**, 1-25.

[18] Kohn, R., Smith, M. and Chan, D. (2001) Nonparametric regression using linear combinations of basis functions. *Statistics and Computing*, **11**, 313-322.

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

[20] Neal, R. (2001) Annealed importance sampling. *Statistics and Computing*, **11**, 125-139.

[21] Pitt, M.K. and Shephard, N. (1999). Filtering via simulation: auxiliary particle filter. *J. Am. Statist. Ass.*, **94**, 590-599.

[22] Robert, C.P. and Casella, G. (1999) *Monte Carlo Statistical Methods*. New York: Springer Verlag.

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