BAYESIAN STATISTICS 8, pp. 1–34.
J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid,
D. Heckerman, A. F. M. Smith and M. West (Eds.)

Sequential Monte Carlo for Bayesian Computation

Pierre Del Moral
Université de Nice, France
delmoral@math.unice.fr

Arnaud Doucet
Univ. British Columbia, Canada
arnaud@cs.ubc.ca

Ajay Jasra
Imperial College London, UK
ajay.jasra@imperial.ac.uk

Summary

Sequential Monte Carlo (SMC) methods are a class of importance sampling and resampling techniques designed to simulate from a sequence of probability distributions. These approaches have become very popular over the last few years to solve sequential Bayesian inference problems (e.g. Doucet et al. 2001). However, in comparison to Markov chain Monte Carlo (MCMC), the application of SMC remains limited when, in fact, such methods are also appropriate in such contexts (e.g. Chopin (2002); Del Moral et al. (2006)). In this paper, we present a simple unifying framework which allows us to extend both the SMC methodology and its range of applications. Additionally, reinterpreting SMC algorithms as an approximation of nonlinear MCMC kernels, we present alternative SMC and iterative self-interacting approximation (Del Moral and Miclo 2004; 2006) schemes. We demonstrate the performance of the SMC methodology on static and sequential Bayesian inference problems.

Keywords and Phrases: Importance Sampling; Nonlinear Markov Chain Monte Carlo; Probit Regression; Sequential Monte Carlo; Stochastic Volatility

1. INTRODUCTION

Consider a sequence of probability measures $\{\pi_n\}_{n\in\mathbb{T}}$ where $\mathbb{T} = \{1, \ldots, P\}$. The distribution $\pi_n(d\mathbf{x}_n)$ is defined on a measurable space $(E_n, \mathcal{E}_n)$. For ease of presentation, we will assume that each $\pi_n(d\mathbf{x}_n)$ admits a density $\pi_n(\mathbf{x}_n)$ with respect to

Pierre Del Moral is Professor of Mathematics at the Université Nice Sophia Antipolis, Arnaud Doucet is Associate Professor of Computer Science and Statistics at the University of British Columbia and Ajay Jasra is Research Fellow in the Department of Mathematics at Imperial College London.

a $\sigma-$finite dominating measure denoted $d\mathbf{x}_n$ and that this density is only known up to a normalizing constant

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

where $\gamma_n : E_n \to \mathbb{R}^+$ is known pointwise, but Z_n might be unknown. 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 S_n the support of π_n, i.e. $S_n = \{\mathbf{x}_n \in E_n : \pi_n(\mathbf{x}_n) > 0\}$.

In this paper, we focus upon sampling from the distributions $\{\pi_n\}_{n\in\mathbb{T}}$ and estimating their normalizing constants $\{Z_n\}_{n\in\mathbb{T}}$ *sequentially*; i.e. first sampling from π_1 and estimating Z_1, then sampling from π_2 and estimating Z_2 and so on. Many computational problems in Bayesian statistics, computer science, physics and applied mathematics can be formulated as sampling from a sequence of probability distributions and estimating their normalizing constants; see for example Del Moral (2004), Iba (2001) or Liu (2001).

1.1. *Motivating Examples*

We now list a few motivating examples.

Optimal filtering for nonlinear non-Gaussian state-space models. Consider an unobserved Markov process $\{X_n\}_{n\geq 1}$ on space $(\mathsf{X}^{\mathbb{N}}, \mathcal{X}^{\mathbb{N}}, \mathbb{P}_\mu)$ where $\mathbb{P}_\mu$ has initial distribution μ and transition density f. The observations $\{Y_n\}_{n\geq 1}$ are assumed to be conditionally independent given $\{X_n\}_{n\geq 1}$ and $Y_n|(X_n = x) \sim g(\cdot|x)$. In this case we define $E_n = \mathsf{X}^n$, $\mathbf{x}_n = x_{1:n}$ ($x_{1:n} \triangleq (x_1, \ldots, x_n)$) and

$$\gamma_n(\mathbf{x}_n) = \mu(x_1)\, g(y_1|x_1) \left\{ \prod_{k=2}^{n} f(x_k|x_{k-1})\, g(y_k|x_k) \right\} \tag{1}$$

This model is appropriate to describe a vast number of practical problems and has been the main application of SMC methods (Doucet et al. 2001). It should be noted that MCMC is not appropriate in such contexts. This is because running P MCMC algorithms, either sequentially (and not using the previous samples in an efficient way) or in parallel is too computationally expensive for large P. Moreover, one often has real-time constraints and thus, in this case, MCMC is not a viable alternative to SMC.

Tempering/annealing. Suppose we are given the problem of simulating from $\pi(\mathbf{x}) \propto \gamma(\mathbf{x})$ defined on E and estimating its normalizing constant $Z = \int_E \gamma(\mathbf{x})\, d\mathbf{x}$. If π is a high-dimensional, non-standard distribution then, to improve the exploration ability of an algorithm, it is attractive to consider an inhomogeneous sequence of P distributions to move "smoothly" from a tractable distribution $\pi_1 = \mu_1$ to the target distribution $\pi_P = \pi$. In this case we have $E_n = E\ \forall n \in \mathbb{T}$ and, for example, we could select a geometric path (Gelman and Meng 1996; Neal 2001)

$$\gamma_n(\mathbf{x}_n) = [\gamma(\mathbf{x}_n)]^{\zeta_n} [\mu_1(\mathbf{x}_n)]^{1-\zeta_n}$$

with $0 \leq \zeta_1 < \cdots < \zeta_P = 1$. Alternatively, to maximize $\pi(\mathbf{x})$, we could consider $\gamma_n(\mathbf{x}_n) = [\gamma(\mathbf{x}_n)]^{\zeta_n}$ where $\{\zeta_n\}$ is such that $0 < \zeta_1 < \cdots < \zeta_P$ and $1 << \zeta_P$ to ensure that $\pi_P(\mathbf{x})$ is concentrated around the set of global maxima of $\pi(\mathbf{x})$. We will demonstrate that it is possible to perform this task using SMC whereas,

typically, one samples from these distributions using either an MCMC kernel of invariant distribution $\pi^*(\mathbf{x}_{1:P}) \propto \gamma_1(\mathbf{x}_1) \times \cdots \times \gamma_P(\mathbf{x}_P)$ (parallel tempering; see Jasra et al. (2005b) for a review) or an inhomogeneous sequence of MCMC kernels (simulated annealing).

Optimal filtering for partially observed point processes. Consider a marked point process $\{c_n, \varepsilon_n\}_{n\geq 1}$ on the real line where c_n is the arrival time of the n^{th} point ($c_n > c_{n-1}$) and ε_n its associated real-valued mark. We assume the marks $\{\varepsilon_n\}$ (resp. the interarrival times $T_n = c_n - c_{n-1}$, $T_1 = c_1 > 0$) are i.i.d. of density f_ε (resp. f_T). We denote by $y_{1:m_t}$ the observations available up to time t and the associated likelihood $g\left(y_{1:m_t} | \{c_n, \varepsilon_n\}_{n\geq 1}\right) = g\left(y_{1:m_t} | c_{1:k_t}, \varepsilon_{1:k_t}\right)$ where $k_t = \arg\max\{i : c_i < t\}$. We are interested in the sequence of posterior distributions at times $\{d_n\}_{n\geq 1}$ where $d_n > d_{n-1}$. In this case, we have $\mathbf{x}_n = \left(k_{d_n}, c_{1:k_{d_n}}, \varepsilon_{1:k_{d_n}}\right)$ and

$$\pi_n(\mathbf{x}_n) \propto g\left(y_{1:m_{d_n}} | k_{d_n}, c_{1:d_n}, \varepsilon_{1:k_{d_n}}\right) \left(\prod_{k=1}^{k_{d_n}} f_\varepsilon(\varepsilon_k) f_T(c_k - c_{k-1})\right) p_{d_n}(k_{d_n} | c_{1:d_n}),$$

where $c_0 = 0$ by convention. These target distributions are all defined on the same space $E_n = E = \biguplus_{k=0}^{\infty} \{k\} \times A_k \times \mathbb{R}^k$ where $A_k = \{c_{1:k} : 0 < c_1 < \cdots < c_k < \infty\}$ but the support S_n of $\pi_n(\mathbf{x}_n)$ is restricted to $\biguplus_{k=0}^{\infty} \{k\} \times A_{k,d_n} \times \mathbb{R}^k$ where $A_{k,d_n} = \{c_{1:k} : 0 < c_1 < \cdots < c_k < d_n\}$, i.e. $S_{n-1} \subset S_n$. This is a sequential, trans-dimensional Bayesian inference problem (see also Del Moral et al. (2006)).

1.2. *Sequential Monte Carlo and Structure of the Article*

SMC methods are a set of simulation-based methods developed to solve the problems listed above, and many more. At a given time n, the basic idea is to obtain a large collection of N weighted random samples $\left\{W_n^{(i)}, \mathbf{X}_n^{(i)}\right\}$ ($i = 1, \ldots, N$, $W_n^{(i)} > 0$; $\sum_{i=1}^N W_n^{(i)} = 1$), $\left\{\mathbf{X}_n^{(i)}\right\}$ being named particles, whose empirical distribution converges asymptotically ($N \to \infty$) to π_n; i.e. for any π_n-integrable function $\varphi : E_n \to \mathbb{R}$

$$\sum_{i=1}^N W_n^{(i)} \varphi\left(\mathbf{X}_n^{(i)}\right) \longrightarrow \int_{E_n} \varphi(\mathbf{x}_n) \pi_n(\mathbf{x}_n) d\mathbf{x}_n \text{ almost surely.}$$

Throughout we will denote $\int_{E_n} \varphi(\mathbf{x}_n) \pi_n(\mathbf{x}_n) d\mathbf{x}_n$ by $\mathbb{E}_{\pi_n}\left(\varphi(\mathbf{X}_n)\right)$. These particles are carried forward over time using a combination of sequential Importance Sampling (IS) and resampling ideas. Broadly speaking, when an approximation $\left\{W_{n-1}^{(i)}, \mathbf{X}_{n-1}^{(i)}\right\}$ of π_{n-1} is available, we seek to move the particles at time n so that they approximate π_n (we will assume that this is not too dissimilar to π_{n-1}), that is, to obtain $\left\{\mathbf{X}_n^{(i)}\right\}$. However, since the $\left\{\mathbf{X}_n^{(i)}\right\}$ are not distributed according to π_n, it is necessary to reweight them with respect to π_n, through IS, to obtain $\left\{W_n^{(i)}\right\}$. In addition, if the variance of the weights is too high (measured through the effective sample size (ESS) (Liu, 2001)), then particles with low weights are eliminated and particles with high weights are multiplied to focus the computational efforts

in "promising" parts of the space. The resampled particles are approximately distributed according to π_n; this approximation improves as $N \to \infty$.

In comparison to MCMC, SMC methods are currently limited, both in terms of their application and framework. In terms of the former, Resample-Move (Chopin 2002; Gilks and Berzuini 2001) is an SMC algorithm which may be used in the same context as MCMC but is not, presumably due to the limited exposure, of applied statisticians, to this algorithm. In terms of the latter, only simple moves have been previously applied to propagate particles, which has serious consequences on the performance of such algorithms. We present here a simple generic mechanism relying on auxiliary variables that allows us to extend the SMC methodology in a principled manner. Moreover, we also reinterpret SMC algorithms as particle approximations of nonlinear and nonhomogeneous MCMC algorithms (Del Moral 2004). This allows us to introduce alternative SMC and iterative self-interacting approximation (Del Moral and Miclo 2004; 2006) schemes. We do not present any theoretical results here but a survey of precise convergence for SMC algorithms can be found in Del Moral (2004) whereas the self-interacting algorithms can be studied using the techniques developed in Del Moral and Miclo (2004; 2006) and Andrieu et al. (2006).

The rest of the paper is organized as follows. Firstly, in Section 2, we review the limitations of the current SMC methodology, present some extensions and describe a generic algorithm to sample from any sequence of distributions $\{\pi_n\}_{n\in\mathbb{T}}$ and estimate $\{Z_n\}_{n\in\mathbb{T}}$ defined in the introduction. Secondly, in Section 3, we reinterpret SMC as an approximation to nonlinear MCMC and discuss an alternative self-interacting approximation. Finally, in Section 4, we present three original applications of our methodology: sequential Bayesian inference for bearings-only tracking (e.g. Gilks and Berzuini (2001)); Bayesian probit regression (e.g. Albert and Chib (1993)) and sequential Bayesian inference for stochastic volatility models (Roberts et al. 2004).

2. SEQUENTIAL MONTE CARLO METHODOLOGY

2.1. *Sequential Importance Sampling*

At time $n-1$, we are interested in estimating π_{n-1} and Z_{n-1}. Let us introduce an importance distribution η_{n-1}. IS is based upon the following identities

$$\begin{array}{l} \pi_{n-1}\left(\mathbf{x}_{n-1}\right) = Z_{n-1}^{-1} w_{n-1}\left(\mathbf{x}_{n-1}\right) \eta_{n-1}\left(\mathbf{x}_{n-1}\right), \\ Z_{n-1} = \int_{E_{n-1}} w_{n-1}(\mathbf{x}_{n-1}) \eta_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1}, \end{array} \tag{2}$$

where the unnormalized importance weight function is equal to

$$w_{n-1}\left(\mathbf{x}_{n-1}\right) = \frac{\gamma_{n-1}\left(\mathbf{x}_{n-1}\right)}{\eta_{n-1}\left(\mathbf{x}_{n-1}\right)}. \tag{3}$$

By sampling N particles $\left\{\mathbf{X}_{n-1}^{(i)}\right\}$ $(i = 1, \ldots, N)$ from η_{n-1} and substituting the empirical measure

$$\eta_{n-1}^{N}(d\mathbf{x}_{n-1}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{X}_{n-1}^{(i)}}(d\mathbf{x}_{n-1})$$

(where δ_x is Dirac measure) to η_{n-1} into (2) we obtain an approximation of π_{n-1} and Z_{n-1} given by

$$\pi_{n-1}^{N}\left(d\mathbf{x}_{n-1}\right) \quad = \sum_{i=1}^{N} W_{n-1}^{(i)} \delta_{X_{n-1}^{(i)}}(d\mathbf{x}_{n-1}), \tag{4}$$

$$Z_{n-1}^N = \frac{1}{N}\sum_{i=1}^N w_{n-1}\left(\mathbf{X}_{n-1}^{(i)}\right), \tag{5}$$

where

$$W_{n-1}^{(i)} = \frac{w_{n-1}\left(\mathbf{X}_{n-1}^{(i)}\right)}{\sum_{j=1}^N w_{n-1}\left(\mathbf{X}_{n-1}^{(j)}\right)}.$$

We now seek to estimate π_n and Z_n. To achieve this we propose to build the importance distribution η_n based upon the current importance distribution η_{n-1} of the particles $\left\{\mathbf{X}_{n-1}^{(i)}\right\}$. We simulate each new particle $\mathbf{X}_n^{(i)}$ according to a Markov kernel $K_n : E_{n-1} \to \mathcal{P}(E_n)$ (where $\mathcal{P}(E_n)$ is the class of probability measures on E_n), i.e. $\mathbf{X}_n^{(i)} \sim K_n\left(\mathbf{x}_{n-1}^{(i)}, \cdot\right)$ so that

$$\eta_n(\mathbf{x}_n) = \eta_{n-1}K_n(\mathbf{x}_n) = \int \eta_{n-1}(d\mathbf{x}_{n-1})\, K_n(\mathbf{x}_{n-1}, \mathbf{x}_n). \tag{6}$$

2.2. *Selection of Transition Kernels*

It is clear that the optimal importance distribution, in the sense of minimizing the variance of (3), is $\eta_n(\mathbf{x}_n) = \pi_n(\mathbf{x}_n)$. Therefore, the optimal transition kernel is simply $K_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \pi_n(\mathbf{x}_n)$. This choice is typically impossible to use (except perhaps at time 1) and we have to formulate sub-optimal choices. We first review conditionally optimal moves and then discuss some alternatives.

2.2.1. *Conditionally optimal moves*

Suppose that we are interested in moving from $\mathbf{x}_{n-1} = (\mathbf{u}_{n-1}, \mathbf{v}_{n-1}) \in E_{n-1} = U_{n-1} \times V_{n-1}$ to $\mathbf{x}_n = (\mathbf{u}_{n-1}, \mathbf{v}_n) \in E_n = U_{n-1} \times V_n$ ($V_n \neq \emptyset$). We adopt the following kernel

$$K_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \mathbb{I}_{\mathbf{u}_{n-1}}(\mathbf{u}_n) q_n(\mathbf{x}_{n-1}, \mathbf{v}_n)$$

where $q_n(\mathbf{x}_{n-1}, \mathbf{v}_n)$ is a probability density of moving from $\mathbf{x}_{n-1}$ to $\mathbf{v}_n$. Consequently, we have

$$\eta_n(\mathbf{x}_n) = \int_{V_{n-1}} \eta_{n-1}(\mathbf{u}_n, d\mathbf{v}_{n-1})\, q_n((\mathbf{u}_n, \mathbf{v}_{n-1}), \mathbf{v}_n).$$

In order to select $q_n(\mathbf{x}_{n-1}, \mathbf{v}_n)$, a sensible strategy consists of using the distribution minimizing the variance of $w_n(\mathbf{x}_n)$ conditional on $\mathbf{u}_{n-1}$. One can easily check that the optimal distribution for this criterion is given by a Gibbs move

$$q_n^{\text{opt}}(\mathbf{x}_{n-1}, \mathbf{v}_n) = \pi_n(\mathbf{v}_n | \mathbf{u}_{n-1})$$

and the associated importance weight satisfies (even if $V_n = \emptyset$)

$$w_n(\mathbf{x}_n) = \frac{\gamma_n(\mathbf{u}_{n-1})}{\eta_{n-1}(\mathbf{u}_{n-1})}. \tag{7}$$

Contrary to the Gibbs sampler, the SMC framework not only requires being able to sample from the full conditional distribution $\pi_n(\mathbf{v}_n | \mathbf{u}_{n-1})$ but also being able to evaluate $\gamma_n(\mathbf{u}_{n-1})$ and $\eta_{n-1}(\mathbf{u}_{n-1})$.

In cases where it is possible to sample from $\pi_n(\mathbf{v}_n|\mathbf{u}_{n-1})$ but impossible to compute $\gamma_n(\mathbf{u}_{n-1})$ and/or $\eta_{n-1}(\mathbf{u}_{n-1})$, we can use an attractive property of IS: we do not need to compute exactly (7), we can use an unbiased estimate of it. We have the identity

$$\gamma_n(\mathbf{u}_{n-1}) = \widehat{\gamma}_n(\mathbf{u}_{n-1}) \int \frac{\gamma_n(\mathbf{u}_{n-1}, \mathbf{v}_n)}{\widehat{\gamma}_n(\mathbf{u}_{n-1}, \mathbf{v}_n)} \widehat{\pi}_n(\mathbf{v}_n|\mathbf{u}_{n-1})\, d\mathbf{v}_n \tag{8}$$

where $\widehat{\gamma}_n(\mathbf{u}_{n-1}, \mathbf{v}_n)$ is selected as an approximation of $\gamma_n(\mathbf{u}_{n-1}, \mathbf{v}_n)$ such that $\int \widehat{\gamma}_n(\mathbf{u}_{n-1}, \mathbf{v}_n)\, d\mathbf{v}_n$ can be computed analytically and it is easy to sample from its associated full conditional $\widehat{\pi}_n(\mathbf{v}_n|\mathbf{u}_{n-1})$. We can calculate an unbiased estimate of $\gamma_n(\mathbf{u}_{n-1})$ using samples from $\widehat{\pi}_n(\mathbf{v}_n|\mathbf{u}_{n-1})$. We also have

$$\frac{1}{\eta_{n-1}(\mathbf{u}_{n-1})} = \frac{1}{\widehat{\eta}_{n-1}(\mathbf{u}_{n-1})} \int \frac{\widehat{\eta}_{n-1}(\mathbf{u}_{n-1}, \mathbf{v}_{n-1})}{\eta_{n-1}(\mathbf{u}_{n-1}, \mathbf{v}_{n-1})} \eta_{n-1}(\mathbf{v}_{n-1}|\mathbf{u}_{n-1})\, d\mathbf{v}_{n-1} \tag{9}$$

where $\widehat{\eta}_{n-1}(\mathbf{u}_{n-1}, \mathbf{v}_{n-1})$ is selected as an approximation of $\eta_{n-1}(\mathbf{u}_{n-1}, \mathbf{v}_{n-1})$ such that $\int \widehat{\eta}_{n-1}(\mathbf{u}_{n-1}, \mathbf{v}_{n-1})\, d\mathbf{v}_{n-1}$ can be computed analytically. So if we can sample from $\eta_{n-1}(\mathbf{v}_{n-1}|\mathbf{u}_{n-1})$, we can calculate an unbiased estimate of (9). This idea has a limited range of applications as in complex cases we do not necessarily have a closed-form expression for $\eta_{n-1}(\mathbf{x}_{n-1})$. However, if one has resampled particles at time $k \leq n-1$, then one has (approximately) $\eta_{n-1}(\mathbf{x}_{n-1}) = \pi_k K_{k+1} K_{k+2} \cdots K_{n-1}(\mathbf{x}_{n-1})$.

2.2.2. *Approximate Gibbs Moves*

In the previous subsection, we have seen that conditionally optimal moves correspond to Gibbs moves. However, in many applications the full conditional distribution $\pi_n(\mathbf{v}_n|\mathbf{u}_{n-1})$ cannot be sampled from. Even if it is possible to sample from it, one might not be able to get a closed-form expression for $\gamma_n(\mathbf{u}_{n-1})$ and we need an approximation $\widehat{\pi}_n(\mathbf{v}_n|\mathbf{u}_{n-1})$ of $\pi_n(\mathbf{v}_n|\mathbf{u}_{n-1})$ to compute an unbiased estimate of it with low variance. Alternatively, we can simply use the following transition kernel

$$K_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \mathbb{I}_{\mathbf{u}_{n-1}}(\mathbf{u}_n)\, \widehat{\pi}_n(\mathbf{v}_n|\mathbf{u}_{n-1}) \tag{10}$$

and the associated importance weight is given by

$$w_n(\mathbf{x}_n) = \frac{\gamma_n(\mathbf{u}_{n-1}, \mathbf{v}_n)}{\eta_{n-1}(\mathbf{u}_{n-1})\, \widehat{\pi}_n(\mathbf{v}_n|\mathbf{u}_{n-1})}. \tag{11}$$

Proceeding this way, we bypass the estimation of $\gamma_n(\mathbf{u}_{n-1})$ which appeared in (7). However, we still need to compute $\eta_{n-1}(\mathbf{u}_{n-1})$ or to obtain an unbiased estimate of its inverse. Unfortunately, this task is very complex except when $\mathbf{u}_{n-1} = \mathbf{x}_{n-1}$(i.e. $V_{n-1} = \emptyset$) in which case we can rewrite (11) as

$$w_n(\mathbf{x}_n) = w_{n-1}(\mathbf{x}_{n-1}) \frac{\gamma_n(\mathbf{x}_{n-1}, \mathbf{v}_n)}{\gamma_n(\mathbf{x}_{n-1})\, \widehat{\pi}_n(\mathbf{v}_n|\mathbf{x}_{n-1})}. \tag{12}$$

This strategy is clearly limited as it can only be used when $E_n = E_{n-1} \times V_n$.

2.2.3. *MCMC and Adaptive moves*

To move from $\mathbf{x}_{n-1} = (\mathbf{u}_{n-1}, \mathbf{v}_{n-1})$ to $\mathbf{x}_n = (\mathbf{u}_{n-1}, \mathbf{v}_n)$ (via K_n), we can adopt an MCMC kernel of invariant distribution $\pi_n(\mathbf{v}_n | \mathbf{u}_{n-1})$. Unlike standard MCMC, there are no (additional) complicated mathematical conditions required to ensure that the usage of adaptive kernels leads to convergence. This is because SMC relies upon IS methodology, that is, we correct for sampling from the wrong distribution via the importance weight. In particular, this allows us to use transition kernels which at time n depends on π_{n-1}, i.e. the "theoretical" transition kernel is of the form $K_{n,\pi_{n-1}}(\mathbf{x}_{n-1}, \mathbf{x}_n)$ and is approximated practically by $K_{n,\pi_{n-1}^N}(\mathbf{x}_{n-1}, \mathbf{x}_n)$. This was proposed and justified theoretically in Crisan and Doucet (2000). An appealing application is described in Chopin (2002) where the variance of $\widehat{\pi}_{n-1}^N$ is used to scale the proposal distribution of an independent MH step of invariant distribution π_n. In Jasra et al. (2005a), one fits a Gaussian mixture model to the particles so as to design efficient trans-dimensional moves in the spirit of Green (2003).

A severe drawback of the strategies mentioned above, is the ability to implement them. This is because we cannot always compute the resulting marginal importance distribution $\eta_n(\mathbf{x}_n)$ given by (6) and, hence, the importance weight $w_n(\mathbf{x}_n)$. In Section 2.3 we discuss how we may solve this problem.

2.2.4. *Mixture of moves*

For complex MCMC problems, one typically uses a combination of MH steps where the parameter components are updated by sub-blocks. Similarly, to sample from high dimensional distributions, a practical SMC sampler will update the components of $\mathbf{x}_n$ via sub-blocks; a mixture of transition kernels can be used at each time n. Let us assume $K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)$ is of the form

$$K_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \sum_{m=1}^{M} \alpha_{n,m}(\mathbf{x}_{n-1}) K_{n,m}(\mathbf{x}_{n-1}, \mathbf{x}_n) \tag{13}$$

where

$$\alpha_{n,m}(\mathbf{x}_{n-1}) \geq 0, \qquad \sum_{m=1}^{M} \alpha_{n,m}(\mathbf{x}_{n-1}) = 1,$$

and $\{K_{n,m}\}$ is a collection of transition kernels. Unfortunately, the direct calculation of the importance weight (6) associated to (13) will be impossible in most cases as $\eta_{n-1}K_{n,m}(\mathbf{x}_n)$ does not admit a closed-form expression. Moreover, even if this were the case, (13) would be expensive to compute pointwise if M is large.

2.2.5. *Summary*

IS, the basis of SMC methods, allows us to consider complex moves including adaptive kernels or non-reversible trans-dimensional moves. In this respect, it is much more flexible than MCMC. However, the major limitation of IS is that it requires the ability to compute the associated importance weights or unbiased estimates of them. In all but simple situations, this is impossible and this severely restricts the application of this methodology. In the following section, we describe a simple auxiliary variable method that allows us to deal with this problem.

2.3. *Auxiliary Backward Markov Kernels*

A simple solution would consist of approximating the importance distribution $\eta_n(\mathbf{x}_n)$ via

$$\eta_{n-1}^N K_n(\mathbf{x}_n) = \frac{1}{N}\sum_{i=1}^N K_n\left(\mathbf{X}_{n-1}^{(i)}, \mathbf{x}_n\right).$$

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 $K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)$ pointwise in important scenarios, e.g. when K_n is an Metropolis-Hastings (MH) kernel of invariant distribution π_n.

We present a simple auxiliary variable idea to deal with this problem (Del Moral et al., 2006). For each forward kernel $K_n : E_{n-1} \to \mathcal{P}(E_n)$, we associate a backward (in time) Markov transition kernel $L_{n-1} : E_n \to \mathcal{P}(E_{n-1})$ and define a new sequence of target distributions $\{\widetilde{\pi}_n(\mathbf{x}_{1:n})\}$ on $E_{1:n} \triangleq E_1 \times \cdots \times E_n$ through

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

where

$$\widetilde{\gamma}_n(\mathbf{x}_{1:n}) = \gamma_n(\mathbf{x}_n)\prod_{k=1}^{n-1} L_k(\mathbf{x}_{k+1}, \mathbf{x}_k).$$

By construction, $\widetilde{\pi}_n(\mathbf{x}_{1:n})$ admits $\pi_n(\mathbf{x}_n)$ as a marginal and Z_n as a normalizing constant. We approximate $\widetilde{\pi}_n(\mathbf{x}_{1:n})$ using IS by using the joint importance distribution

$$\eta_n(\mathbf{x}_{1:n}) = \eta_1(\mathbf{x}_1)\prod_{k=2}^{n} K_k(\mathbf{x}_{k-1}, \mathbf{x}_k).$$

The associated importance weight satisfies

$$\begin{aligned} w_n(\mathbf{x}_{1:n}) &= \frac{\widetilde{\gamma}_n(\mathbf{x}_{1:n})}{\eta_n(\mathbf{x}_{1:n})} \\ &= w_{n-1}(\mathbf{x}_{1:n-1})\,\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n). \end{aligned} \tag{14}$$

where the incremental importance weight $\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n)$ is given by

$$\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \frac{\gamma_n(\mathbf{x}_n)\, L_{n-1}(\mathbf{x}_n, \mathbf{x}_{n-1})}{\gamma_{n-1}(\mathbf{x}_{n-1})\, K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)}.$$

Given that this Radon-Nikodym derivative is well-defined, the method will produce asymptotically ($N \to \infty$) consistent estimates of $\mathbb{E}_{\widetilde{\pi}_n}\left(\varphi(\mathbf{X}_{1:n})\right)$ and Z_n. However, the performance of the algorithm will be dependent upon the choice of the kernels $\{L_k\}$.

2.3.1. *Optimal backward kernels*

Del Moral et al. (2006) establish that the backward kernels which minimize the variance of the importance weights, $w_n(\mathbf{x}_{1:n})$, are given by

$$L_k^{\text{opt}}(\mathbf{x}_{k+1}, \mathbf{x}_k) = \frac{\eta_k(\mathbf{x}_k)\, K_{k+1}(\mathbf{x}_k, \mathbf{x}_{k+1})}{\eta_{k+1}(\mathbf{x}_{k+1})} \tag{15}$$

for $k = 1, ..., n-1$. This can be verified easily by noting that

$$\eta_n(\mathbf{x}_{1:n}) = \eta_n(\mathbf{x}_n)\prod_{k=1}^{n-1} L_k^{\text{opt}}(\mathbf{x}_{k+1}, \mathbf{x}_k).$$

It is typically impossible, in practice, to use these optimal backward kernels as they rely on marginal distributions which do not admit any closed-form expression. However, this suggests that we should select them as an approximation to (15). The key point is that, even if they are different from (15), the algorithm will still provide asymptotically consistent estimates.

Compared to a "theoretical" algorithm computing the weights (3), the price to pay for avoiding to compute $\eta_n(\mathbf{x}_n)$ (i.e. not using $L_k^{\text{opt}}(\mathbf{x}_{k+1}, \mathbf{x}_k)$) is that the variance of the Monte Carlo estimates based upon (14) will be larger. For example, even if we set $\pi_n(\mathbf{x}_n) = \pi(\mathbf{x}_n)$ and $K_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = K(\mathbf{x}_{n-1}, \mathbf{x}_n)$ is an ergodic MCMC kernel of invariant distribution π then the variance of $w_n(\mathbf{x}_{1:n})$ will fail to stabilize (or become infinite in some cases) over time for any backward kernel $L_k(\mathbf{x}_{k+1}, \mathbf{x}_k) \neq L_k^{\text{opt}}(\mathbf{x}_{k+1}, \mathbf{x}_k)$ whereas the variance of (3) will decrease towards zero. The resampling step in SMC will deal with this problem by resetting the weights when their variance is too high.

At time n, the backward kernels $\{L_k(\mathbf{x}_{k+1}, \mathbf{x}_k)\}$ for $k = 1, ..., n-2$ have already been selected and we are interested in some approximations of $L_{n-1}^{\text{opt}}(\mathbf{x}_n, \mathbf{x}_{n-1})$ controlling the evolution of the variance of $w_n(\mathbf{x}_{1:n})$.

2.3.2. *Suboptimal backward kernels*

• *Substituting π_{n-1} for η_{n-1}.* Equation (15) suggests that a sensible sub-optimal strategy consists of substituting π_{n-1} for η_{n-1} to obtain

$$L_{n-1}(\mathbf{x}_n, \mathbf{x}_{n-1}) = \frac{\pi_{n-1}(\mathbf{x}_{n-1}) K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)}{\pi_{n-1} K_n(\mathbf{x}_n)} \tag{16}$$

which yields

$$\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \frac{\gamma_n(\mathbf{x}_n)}{\int \gamma_{n-1}(d\mathbf{x}_{n-1}) K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)}. \tag{17}$$

It is often more convenient to use (17) than (15) 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 (15) is equal to (16).

• *Gibbs and Approximate Gibbs Moves.* Consider the conditionally optimal move described earlier where

$$K_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \mathbb{I}_{\mathbf{u}_{n-1}}(\mathbf{u}_n) \pi_n(\mathbf{v}_n | \mathbf{u}_{n-1}) \tag{18}$$

In this case (16) and (17) are given by

$$L_{n-1}(\mathbf{x}_n, \mathbf{x}_{n-1}) = \mathbb{I}_{\mathbf{u}_n}(\mathbf{u}_{n-1}) \pi_{n-1}(\mathbf{v}_{n-1} | \mathbf{u}_{n-1}),$$

$$\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \frac{\gamma_n(\mathbf{u}_{n-1})}{\gamma_{n-1}(\mathbf{u}_{n-1})}.$$

An unbiased estimate of $\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n)$ can also be computed using the techniques described in 2.2.1. When it is impossible to sample from $\pi_n(\mathbf{v}_n | \mathbf{u}_{n-1})$ and/or compute $\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n)$, we may be able to construct an approximation $\widehat{\pi}_n(\mathbf{v}_n | \mathbf{u}_{n-1})$ of $\pi_n(\mathbf{v}_n | \mathbf{u}_{n-1})$ to sample the particles and another approximation $\widehat{\pi}_{n-1}(\mathbf{v}_{n-1} | \mathbf{u}_{n-1})$ of $\pi_{n-1}(\mathbf{v}_{n-1} | \mathbf{u}_{n-1})$ to obtain

$$L_{n-1}(\mathbf{x}_n, \mathbf{x}_{n-1}) = \mathbb{I}_{\mathbf{u}_n}(\mathbf{u}_{n-1}) \widehat{\pi}_{n-1}(\mathbf{v}_{n-1} | \mathbf{u}_{n-1}), \tag{19}$$

$$\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \frac{\gamma_n(\mathbf{u}_{n-1}, \mathbf{v}_n)\,\widehat{\pi}_{n-1}(\mathbf{v}_{n-1}|\,\mathbf{u}_{n-1})}{\gamma_{n-1}(\mathbf{u}_{n-1}, \mathbf{v}_{n-1})\,\widehat{\pi}_n(\mathbf{v}_n|\,\mathbf{u}_{n-1})}. \tag{20}$$

• *MCMC Kernels.* A generic alternative approximation of (16) can also be made when K_n is an MCMC kernel of invariant distribution π_n. This has been proposed explicitly in (Jarzynski (1997), Neal (2001)) and implicitly in all papers introducing MCMC moves within SMC, e.g. Chopin (2002), Gilks and Berzuini (2001). It is given by

$$L_{n-1}(\mathbf{x}_n, \mathbf{x}_{n-1}) = \frac{\pi_n(\mathbf{x}_{n-1})\,K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)}{\pi_n(\mathbf{x}_n)} \tag{21}$$

and will be a good approximation of (16) if $\pi_{n-1} \approx \pi_n$; note that (21) is the reversal Markov kernel associated with K_n. In this case, the incremental weight satisfies

$$\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \frac{\gamma_n(\mathbf{x}_{n-1})}{\gamma_{n-1}(\mathbf{x}_{n-1})}. \tag{22}$$

This expression (22) is remarkable as it is easy to compute and valid *irrespective* of the MCMC kernel adopted. It is also counter-intuitive: if $K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)$ is mixing quickly so that, approximately, $\mathbf{X}_n^{(i)} \sim \pi_n$ then the particles would still be weighted. The use of resampling helps to mitigate this problem; see (Del Moral et al. 2006, Section 3.5) for a detailed discussion.

Contrary to (16), this approach does not apply in scenarios where $E_{n-1} = E_n$ but $S_{n-1} \subset S_n$ as discussed in Section 1 (optimal filtering for partially observed processes). Indeed, in this case

$$L_{n-1}(\mathbf{x}_n, \mathbf{x}_{n-1}) = \frac{\pi_n(\mathbf{x}_{n-1})\,K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)}{\int_{S_{n-1}} \pi_n(\mathbf{x}_{n-1})\,K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)\,d\mathbf{x}_{n-1}} \tag{23}$$

but the denominator of this expression is different from $\pi_n(\mathbf{x}_n)$ as the integration is over S_{n-1} and not S_n.

2.3.3. *Mixture of Markov Kernels*

When the transition kernel is given by a mixture of M moves as in (13), one should select $L_{n-1}(\mathbf{x}_n, \mathbf{x}_{n-1})$ as a mixture

$$L_{n-1}(\mathbf{x}_n, \mathbf{x}_{n-1}) = \sum_{m=1}^{M} \beta_{n-1,m}(\mathbf{x}_n)\,L_{n-1,m}(\mathbf{x}_n, \mathbf{x}_{n-1}) \tag{24}$$

where $\beta_{n-1,m}(\mathbf{x}_n) \geq 0$, $\sum_{m=1}^{M} \beta_{n-1,m}(\mathbf{x}_n) = 1$ and $\{L_{n-1,m}\}$ is a collection of backward transition kernels. Using (15), it is indeed easy to show that the optimal backward kernel corresponds to

$$\begin{aligned} \beta_{n-1,m}^{\mathrm{opt}}(\mathbf{x}_n) &\propto \int \alpha_{n,m}(\mathbf{x}_{n-1})\,\eta_{n-1}(\mathbf{x}_{n-1})\,K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)\,d\mathbf{x}_{n-1}, \\ L_{n-1,m}^{\mathrm{opt}}(\mathbf{x}_n, \mathbf{x}_{n-1}) &= \frac{\alpha_{n,m}(\mathbf{x}_{n-1})\eta_{n-1}(\mathbf{x}_{n-1})K_n(\mathbf{x}_{n-1}, \mathbf{x}_{nn})}{\int \alpha_{n,m}(\mathbf{x}_{n-1})\eta_{n-1}(\mathbf{x}_{n-1})K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)d\mathbf{x}_{n-1}}. \end{aligned}$$

Various approximations to $\beta_{n-1,m}^{\mathrm{opt}}(\mathbf{x}_n)$ and $L_{n-1,m}^{\mathrm{opt}}(\mathbf{x}_n, \mathbf{x}_{n-1})$ have to be made in practice.

Moreover, to avoid computing a sum of M terms, we can introduce a discrete latent variable $M_n \in \mathcal{M}$, $\mathcal{M} = \{1, \ldots, M\}$ such that $\mathbb{P}(M_n = m) = \alpha_{n,m}(\mathbf{x}_{n-1})$ and perform IS on the extended space. This yields an incremental importance weight equal to

$$\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n, m_n) = \frac{\gamma_n(\mathbf{x}_n)\beta_{n-1,m_n}(\mathbf{x}_n) L_{n-1,m_n}(\mathbf{x}_n, \mathbf{x}_{n-1})}{\gamma_{n-1}(\mathbf{x}_{n-1})\alpha_{n,m_n}(\mathbf{x}_{n-1}) K_{n,m_n}(\mathbf{x}_{n-1}, \mathbf{x}_n)}.$$

2.4. *A Generic SMC Algorithm*

We now describe a generic SMC algorithm to approximate the sequence of targets $\{\pi_n\}$ based on kernel K_n; the extension to mixture of moves being straightforward. The particle representation is resampled using an (unbiased) systematic resampling scheme whenever the ESS at time n given by $\left[\sum_{i=1}^N (W_n^{(i)})^2\right]^{-1}$ is below a prespecified threshold, say $N/2$ (Liu, 2001).

- **At time** $n = 1$. *Sample* $\mathbf{X}_1^{(i)} \sim \eta_1$ *and compute* $W_1^{(i)} \propto w_1\left(\mathbf{X}_1^{(i)}\right)$.
  If ESS $<$ Threshold, resample the particle representation $\left\{W_1^{(i)}, \mathbf{X}_1^{(i)}\right\}$.
- **At time** n; $n \geq 2$. *Sample* $\mathbf{X}_n^{(i)} \sim K_n\left(\mathbf{X}_{n-1}^{(i)}, \cdot\right)$ *and compute* $W_n^{(i)} \propto W_{n-1}^{(i)} \widetilde{w}_n\left(\mathbf{X}_{n-1}^{(i)}, \mathbf{X}_n^{(i)}\right)$.
  If ESS $<$ Threshold, resample the particle representation $\left\{W_n^{(i)}, \mathbf{X}_n^{(i)}\right\}$.
  The target π_n is approximated through

$$\pi_n^N(d\mathbf{x}_n) = \sum_{i=1}^N W_n^{(i)} \delta_{\mathbf{X}_n^{(i)}}(d\mathbf{x}_n).$$

In addition, the approximation $\left\{W_{n-1}^{(i)}, \mathbf{X}_n^{(i)}\right\}$ of $\pi_{n-1}(\mathbf{x}_{n-1}) K_n(\mathbf{x}_{n-1}, \mathbf{x}_n)$ obtained after the sampling step allows us to approximate

$$\frac{Z_n}{Z_{n-1}} = \frac{\int \gamma_n(\mathbf{x}_n)\, d\mathbf{x}_n}{\int \gamma_{n-1}(\mathbf{x}_{n-1})\, d\mathbf{x}_{n-1}} \text{ by } \widehat{\frac{Z_n}{Z_{n-1}}} = \sum_{i=1}^N W_{n-1}^{(i)} \widetilde{w}_n\left(\mathbf{X}_{n-1}^{(i)}, \mathbf{X}_n^{(i)}\right). \tag{25}$$

Alternatively, it is possible to use path sampling (Gelman and Meng, 1998) to compute this ratio.

3. NONLINEAR MCMC, SMC AND SELF-INTERACTING APPROXIMATIONS

For standard Markov chains, the transition kernel, say Q_n, is a linear operator in the space of probability measures, *i.e.*, we have $\mathbf{X}_n \sim Q_n(\mathbf{X}_{n-1}, \cdot)$ and the distribution μ_n of $\mathbf{X}_n$ satisfies $\mu_n = \mu_{n-1} Q_n$. Nonlinear Markov chains are such that $\mathbf{X}_n \sim Q_{\mu_{n-1},n}(\mathbf{X}_{n-1}, \cdot)$, i.e. the transition of $\mathbf{X}_n$ depends not only on $\mathbf{X}_{n-1}$ but also on μ_{n-1} and we have

$$\mu_n = \mu_{n-1} Q_{n,\mu_{n-1}}. \tag{26}$$

In a similar fashion to MCMC, it is possible to design nonlinear Markov chain kernels admitting a fixed target π (Del Moral and Doucet 2003). Such a procedure is attractive as one can design nonlinear kernels with theoretically better mixing properties than linear kernels. Unfortunately, it is often impossible to simulate exactly such nonlinear Markov chains as we do not have a closed-form expression for μ_{n-1}. We now describe a general collection of nonlinear kernels and how to produce approximations of them.

3.1. *Nonlinear MCMC Kernels to Simulate from a Sequence of Distributions*

Suppose that we can construct a collection of nonlinear Markov kernels such that

$$\widetilde{\pi}_n = \widetilde{\pi}_{n-1} Q_{n,\widetilde{\pi}_{n-1}}$$

where $\{\widetilde{\pi}_n\}$ is the sequence of auxiliary target distributions (on $(E_{1:n}, \mathcal{E}_{1:n})$) associated to $\{\pi_n\}$ and $Q_{n,\mu} : \mathcal{P}(E_{1:n-1}) \times E_{n-1} \to \mathcal{P}(E_{1:n})$. The simplest transition kernel is given by

$$Q_{n,\mu}\left(\mathbf{x}_{1:n-1}, \mathbf{x}'_{1:n}\right) = \Psi_n\left(\mu \times K_n\right)\left(\mathbf{x}'_{1:n}\right) \tag{27}$$

where $\Psi_n : \mathcal{P}(E_{1:n}) \to \mathcal{P}(E_{1:n})$

$$\Psi_n(\nu)\left(d\mathbf{x}'_{1:n}\right) = \frac{\nu\left(d\mathbf{x}'_{1:n}\right)\widetilde{w}_n\left(\mathbf{x}'_{n-1}, \mathbf{x}'_n\right)}{\int \nu\left(d\mathbf{x}_{1:n}\right)\widetilde{w}_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right)}\,.$$

is a Boltzmann-Gibbs distribution.

If $\widetilde{w}_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right) \leq C_n$ for any $(\mathbf{x}_{n-1}, \mathbf{x}_n)$, we can also consider an alternative kernel given by

$$\begin{aligned} Q_{n,\mu}(\mathbf{x}_{1:n-1}, d\mathbf{x}'_{1:n}) = {} & K_n(\mathbf{x}_{n-1}, d\mathbf{x}'_n)\frac{\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}'_n)}{C_n}\delta_{\mathbf{x}_{1:n-1}}\left(d\mathbf{x}'_{1:n-1}\right) \\ + {} & \left(1 - \int_{E_{1:n}} K_n(\mathbf{x}_{n-1}, d\mathbf{x}'_n)\frac{\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}'_n)}{C_n}\delta_{\mathbf{x}_{1:n-1}}\left(d\mathbf{x}'_{1:n-1}\right)\right) \\ \times {} & \Psi_n(\mu \times K_n)(d\mathbf{x}'_{1:n}). \end{aligned} \tag{28}$$

This algorithm can be interpreted as a nonlinear version of the MH algorithm. Given $\mathbf{x}_{1:n-1}$ we sample $\mathbf{x}'_n \sim K_n(\mathbf{x}_{n-1}, \cdot)$ and with probability $\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}'_n)/C_n$ we let $\mathbf{x}'_{1:n} = (\mathbf{x}_{1:n-1}, \mathbf{x}'_n)$, otherwise we sample a new $\mathbf{x}'_{1:n}$ from the Boltzmann-Gibbs distribution.

3.2. *SMC and Self-Interacting Approximations*

In order to simulate the nonlinear kernel, we need to approximate (26) given here by (27) or (28). The SMC algorithm described in Section 2 can be interpreted as a simple Monte Carlo implementation of (27). Whenever $\widetilde{w}_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right) \leq C_n$, it is also possible to approximate (28) instead. Under regularity assumptions, it can be shown that this alternative Monte Carlo approximation has a lower asymptotic variance than (27) if multinomial resampling is used to sample from the Boltzmann-Gibbs distribution (Chapter 9 of Del Moral, 2004).

In cases where one does not have real-time constraints and the number P of target distributions $\{\pi_n\}$ is fixed it is possible to develop an alternative iterative approach. The idea consists of initializing the algorithm with some Monte Carlo estimates $\left\{\widetilde{\pi}_n^{N_0}\right\}$ of the targets consisting of empirical measures (that is $\frac{1}{N_0}\sum_{i=1}^{N_0}\delta_{X_{n,1:n}^{(i)}}$) of N_0 samples. For the sake of simplicity, we assume it is possible to sample exactly from $\widetilde{\pi}_1 = \pi_1$. Then the algorithm proceeds as follows at iteration i; the first iteration being indexed by $i = N_0 + 1$.

- **At time** $n = 1$. *Sample* $\mathbf{X}_{1,1}^{(i)} \sim \widetilde{\pi}_1$ *and set* $\widetilde{\pi}_1^i = \left(1 - \frac{1}{i}\right)\widetilde{\pi}_1^{i-1} + \frac{1}{i}\delta_{\mathbf{X}_{1,1}^{(i)}}$.
- **At time** n; $n = 2, ..., P$. *Sample* $\mathbf{X}_{n,1:n}^{(i)} \sim Q_{n,\widetilde{\pi}_{n-1}^i}\left(\mathbf{X}_{n-1,1:n-1}^{(i)}, \cdot\right)$ *and set* $\widetilde{\pi}_n^i = \left(1 - \frac{1}{i}\right)\widetilde{\pi}_n^{i-1} + \frac{1}{i}\delta_{\mathbf{X}_{n,1:n}^{(i)}}$.

In practice, we are interested only in $\{\pi_n\}$ and not $\{\widetilde{\pi}_n\}$ so we only need to store at time n the samples $\left\{\mathbf{X}_{n,n-1:n}^{(i)}\right\}$ asymptotically distributed according to $\pi_n(x_n) L_{n-1}(x_n, x_{n-1})$. We note that such stochastic processes, described above, are *self-interacting*; see Del Moral and Miclo (2004; 2006), Andrieu *et al.* (2006) and Brockwell and Doucet (2006) in the context of Monte Carlo simulation.

4. APPLICATIONS

4.1. *Block Sampling for Optimal Filtering*

4.1.1. *SMC Sampler*

We consider the class of nonlinear non-Gaussian state-space models discussed in Section 1. In this case the sequence of target distribution defined on $E_n = \mathsf{X}^n$ is given by (1). In the context where one has real-time constraints, we need to design a transition kernel K_n which updates only a fixed number of components of $\mathbf{x}_n$ to maintain a computational complexity independent of n.

The standard approach consists of moving from $\mathbf{x}_{n-1} = \mathbf{u}_{n-1}$ to $\mathbf{x}_n = (\mathbf{x}_{n-1}, x_n) = (\mathbf{u}_{n-1}, \mathbf{v}_n)$ using

$$\pi_n(\mathbf{v}_n | \mathbf{u}_{n-1}) = p(x_n | y_n, x_{n-1}) \propto f(x_n | x_{n-1})\, g(y_n | x_n).$$

This distribution is often referred to (abusively) as the optimal importance distribution in the literature, e.g. Doucet et al. (2001); this should be understood as optimal *conditional upon* $\mathbf{x}_{n-1}$. In this case we can rewrite (7) as

$$w_n(\mathbf{x}_n) = w_{n-1}(\mathbf{x}_{n-1})\, p(y_n | x_{n-1}) \propto w_{n-1}(\mathbf{x}_{n-1})\, \frac{p(\mathbf{x}_{n-1} | y_{1:n})}{p(\mathbf{x}_{n-1} | y_{1:n-1})} \tag{29}$$

If one can sample from $p(x_n | y_n, x_{n-1})$ but cannot compute (29) in closed-form then we can obtain an unbiased estimate of it using an easy to sample distribution approximating it

$$\widehat{\pi}_n(\mathbf{v}_n | \mathbf{u}_{n-1}) = \widehat{p}(x_n | y_n, x_{n-1}) = \frac{\widehat{f}(x_n | x_{n-1})\, \widehat{g}(y_n | x_n)}{\int \widehat{f}(x_n | x_{n-1})\, \widehat{g}(y_n | x_n)\, dx_n}$$

and the identity

$$\begin{aligned} p(y_n | x_{n-1}) &= \int \widehat{f}(x_n | x_{n-1})\, \widehat{g}(y_n | x_n)\, dx_n \\ &\quad \times \int \frac{f(x_n | x_{n-1}) g(y_n | x_n)}{\widehat{f}(x_n | x_{n-1}) \widehat{g}(y_n | x_n)} \widehat{p}(x_n | y_n, x_{n-1})\, dx_n. \end{aligned}$$

An alternative consists of moving using $\widehat{p}(x_n|y_n, x_{n-1})$ -see (10)- and computing the weights using (12)

$$w_n(\mathbf{x}_n) = w_{n-1}(\mathbf{x}_{n-1}) \frac{f(x_n|x_{n-1})\, g(y_n|x_n)}{\widehat{p}(x_n|y_n, x_{n-1})}$$

We want to emphasize that such sampling strategies can perform poorly even if one can sample from $p(x_n|y_n, x_{n-1})$ and compute exactly the associated importance weight. Indeed, in situations where the discrepancy between $p(\mathbf{x}_{n-1}|y_{1:n-1})$ and $p(\mathbf{x}_{n-1}|y_{1:n})$ is high, then the weights (29) will have a large variance. An alternative strategy consists not only of sampling X_n at time n but also of updating the block of variables $X_{n-R+1:n-1}$ where $R > 1$. In this case we seek to move from $\mathbf{x}_{n-1} = (\mathbf{u}_{n-1}, \mathbf{v}_{n-1}) = (x_{1:n-R}, x_{n-R+1:n-1})$ to $\mathbf{x}_n = (\mathbf{u}_{n-1}, \mathbf{v}_n) = (x_{1:n-R}, x'_{n-R+1:n})$ and the conditionally optimal distribution is given by

$$\pi_n(\mathbf{v}_n|\mathbf{u}_{n-1}) = p\left(x'_{n-R+1:n}\middle|y_{n-R+1:n}, x_{n-R}\right).$$

Although attractive, this strategy is difficult to apply, as sampling from $p(x'_{n-R+1:n}|y_{n-R+1:n}, x_{n-R})$ becomes more difficult as R increases. Moreover, it requires the ability to compute or obtain unbiased estimates of both $p(y_{n-R+1:n}|x_{n-R})$ and $1/\eta_{n-1}(x_{1:n-R})$ to calculate (7). If we use an approximation $\widehat{\pi}_n(\mathbf{v}_n|\mathbf{u}_{n-1})$ of $\pi_n(\mathbf{v}_n|\mathbf{u}_{n-1})$ to move the particles, it remains difficult to compute (11) as we still require an unbiased estimate of $1/\eta_{n-1}(x_{1:n-R})$. The discussion of Section 2.3.2 indicates that, alternatively, we can simply weight the particles sampled using $\widehat{\pi}_n(\mathbf{v}_n|\mathbf{u}_{n-1})$ by (20); this only requires us being able to derive an approximation of $\pi_{n-1}(\mathbf{v}_{n-1}|\mathbf{u}_{n-1})$.

4.1.2. *Model and Simulation details*

We now present numerical results for a bearings-only-tracking example (Gilks and Berzuini, 2001). The target is modelled using a standard constant velocity model

$$X_n = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} X_{n-1} + V_n,$$

with V_n i.i.d. $\mathcal{N}_4(0, \Sigma)$ ($\mathcal{N}_r(a, b)$ is the $r-$dimensional normal distribution with mean a and covariance b) and

$$\Sigma = 5 \begin{pmatrix} 1/3 & 1/2 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 0 & 0 & 1/3 & 1/2 \\ 0 & 0 & 1/2 & 1 \end{pmatrix}.$$

The state vector $X_n = \left(X_n^1, X_n^2, X_n^3, X_n^4\right)^T$ is such that X_n^1 (resp. X_n^3) corresponds to the horizontal (resp. vertical) position of the target whereas X_n^2 (resp. X_n^4) corresponds to the horizontal (resp. vertical) velocity. One only receives observations of the bearings of the target

$$Y_n = \tan^{-1}\left(\frac{X_n^3}{X_n^1}\right) + W_n$$

where W_n is i.i.d. $\mathcal{N}\left(0, 10^{-4}\right)$; i.e. the observations are almost noiseless. This is representative of real-world tracking scenarios.

We build an approximation $\widehat{\pi}_n\left(\mathbf{v}_n|\,\mathbf{u}_{n-1}\right)$ (respectively $\widehat{\pi}_{n-1}\left(\mathbf{v}_{n-1}|\,\mathbf{u}_{n-1}\right)$) of $\pi_n\left(\mathbf{v}_n|\,\mathbf{u}_{n-1}\right)$ (respectively $\widehat{\pi}_{n-1}\left(\mathbf{v}_{n-1}|\,\mathbf{u}_{n-1}\right)$) using the forward-backward sampling formula for a linear Gaussian approximation of the model based on the Extended Kalman Filter (EKF); see Doucet et al. (2006) for details. We compare

• The block sampling SMC algorithms denoted SMC(R) for $R = 1, 2, 5$ and 10 which are using the EKF proposal.

• Two Resample-Move algorithms as described in (Gilks and Berzuini, 2001), where the SMC(1) is used followed by: (i) one at a time MH moves using an approximation of $p\left(x_k|\,y_k, x_{k-1}, x_{k+1}\right)$ as a proposal (RML(10)) over a lag $L = 10$; and (ii) using the EKF proposal for $L = 10$ (RMFL(10)). The acceptance probabilities of those moves were in all cases between (0.5,0.6).

Systematic resampling is performed whenever the ESS goes below $N/2$. The results are displayed in Table 1.

Table 1: *Average number of resampling steps for 100 simulations, 100 time instances per simulations using $N = 1000$ particles.*

Filter	# Time Resampled
SMC(1)	44.6
RML(10)	45.2
RMFL(10)	43.3
SMC(2)	34.9
SMC(5)	4.6
SMC(10)	1.3

The standard algorithms -namely, SMC(1), RML(10) and RMFL(10) - need to resample very often as the ESS drop below $N/2$; see the 2^{nd} column of Table 1. In particular, the Resample-Move algorithms resample as much as SMC(1) despite their computational complexity being similar to SMC(10); this is because MCMC steps are only introduced after an SMC(1) step has been performed. Conversely, as R increases, the number of resampling steps required by SMC(R) methods decreases dramatically. Consequently, the number of unique particles $\left\{X_1^{(i)}\right\}$ approximating the final target $p\left(x_1|\,y_{1:100}\right)$ remains large whereas it is close to 1 for standard methods.

4.2. *Binary Probit Regression*

Our second application, related to the tempering example in Section 1, is the Bayesian binary regression model in (for example) Albert and Chib (1993). The analysis of binary data via generalized linear models often occurs in applied Bayesian statistics and the most commonly used technique to draw inference is the auxiliary variable Gibbs sampler (Albert and Chib 1993). It is well known (e.g. Holmes and Held 2006) that such a simulation method can perform poorly, due to the strong posterior dependency between the regression and auxiliary variables. In this example we illustrate that SMC samplers can provide significant improvements over the auxiliary variable Gibbs sampler with little extra coding effort and comparable CPU times. Further, we demonstrate that the SMC algorithm based on (18)

can greatly improve the performance of Resample-Move (Chopin, 2002; Gilks and Berzuini, 2001) based on (21).

4.2.1. *Model*

The model assumes that we observe binary data $Y_1, \ldots, Y_u$, with associated $r-$dimensional covariates $X_1, \ldots, X_u$ and that the Y_i, $i = 1, \ldots, u$ are i.i.d.:

$$Y_i|\beta \quad \sim \mathcal{B}(\Phi(x_i'\beta))$$

where $\mathcal{B}$ is the Bernoulli distribution, β is a $r-$dimensional vector and Φ is the standard normal CDF. We denote by x the $u \times r$ design matrix (we do not consider models with an intercept).

Albert and Chib (1993) introduced an auxiliary variable Z_i to facilitate application of the Gibbs sampler. That is, we have:

$$\begin{aligned} Y_i|Z_i &= \begin{cases} 1 & \text{if } Z_i > 0 \\ 0 & \text{otherwise} \end{cases} \\ Z_i &= x_i'\beta + \epsilon_i \\ \epsilon_i &\sim \mathcal{N}(0,1). \end{aligned}$$

In addition, we assume $\beta \sim \mathcal{N}_r(b, v)$. Standard manipulations establish that the marginal posterior $\pi(\beta|y_{1:u}, x_{1:u})$ concides with that of the original model.

4.2.2. *Performance of the MCMC algorithm*

To illustrate that MCMC-based inference for binary probit regression does not always perform well, we consider the following example. We simulated 200 data points, with $r = 20$ covariates. We set the priors as $b = 0$ and $v = \text{diag}(100)$. Recall that the Gibbs sampler of Albert and Chib (1993) generates from full conditionals:

$$\begin{aligned} \beta|\cdots &\sim \mathcal{N}_r(B, V) \\ B &= V(v^{-1}b + x'z) \\ V &= (v^{-1} + x'x)^{-1} \\ \pi(z_i|\cdots) &\sim \begin{cases} \phi(z_i; x_i'\beta, 1)\mathbb{I}_{\{z_i>0\}}(z_i) & \text{if } y_i = 1 \\ \phi(z_i; x_i'\beta, 1)\mathbb{I}_{\{z_i\leq 0\}}(z_i) & \text{otherwise} \end{cases} \end{aligned}$$

where $|\cdots$ denotes conditioning on all other random variables in the model and $\phi(\cdot)$ is the normal density. It should be noted that there are more advanced MCMC methods for these class of models (e.g. Holmes and Held (2006)), but we only consider the method of Albert and Chib (1993) as it forms a building block of the SMC sampler below. We ran the MCMC sampler for 100000 iterations, thinning the samples to every 100. The CPU time was approximately 421 seconds.

In Figure 1 (top row) we can observe two of the traces of the twenty sampled regression coefficients. These plots indicate very slow mixing, due to the clear autocorrelations and the thinning of the Markov chain. Whilst we might run the sampler for an excessive period of time (that is, enough to substantially reduce the autocorrelations of the samples), it is preferable to construct an alternative simulation procedure. This is to ensure that we are representing all of the regions of high posterior probability that may not occur using this MCMC sampler.

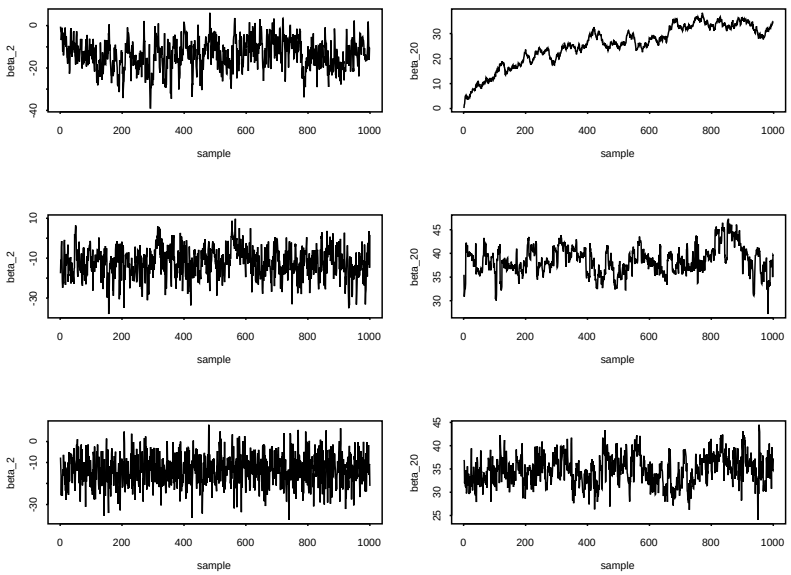

Figure 1: *Sampled coefficients from the binary regression example. For the MCMC (top row), we ran the Gibbs sampler of Albert and Chib (1993) for 100000 iterations and stored every 100th (CPU time 421 sec). For the reversal SMC (middle row) we ran 1000 particles for 200 time steps (CPU 681 sec). For the Gibbs SMC (bottom row) we did the same except the CPU was 677.*

4.2.3. *SMC Sampler*

We now develop an SMC approach to draw inference from the binary logistic model. We consider a sequence of densities induced by the following error at time n:

$$\epsilon_i \quad \sim \mathcal{N}(0, \zeta_n).$$

with $1 < \zeta_1 > \cdots > \zeta_P = 1$.

To sample the particles, we adopt the MCMC kernel above, associated to the density at time n. At time n we sample new $z_{1:u}, \beta$ from:

$$K_n((z_{1:u}, \beta), (z'_{1:u}, \beta')) = \pi_n(z'_{1:u}|\beta, y_{1:u}, x_{1:u})\mathbb{I}_\beta(\beta').$$

We then sample β from the full conditional (since this kernel admits π_n as an invariant measure we can adopt backward kernel (21) and so the incremental weight is 1). For the corresponding backward kernel, L_{n-1}, we consider two options (21) and (18). Since (18) is closer to the optimal kernel, we would expect that the performance under the second kernel to be better than the first (in terms of weight degeneracy).

4.2.4. *Performance of SMC Sampler*

We ran the two SMC samplers above for 50, 100 and 200 time points. We sampled 1000 particles and resampled upon the basis of the ESS dropping to $N/2$ using systematic resampling. The initial importance distribution was a multivariate normal centered at a point simulated from an MCMC sampler and the full conditional density for $z_{1:u}$. We found that this performed noticeably better than using the prior for β.

It should be noted that we did not have to store N, $u-$dimensional vectors. This is possible due to the fact that we can simulate from $\pi_n(z_{1:u}|\cdots)$ and that the incremental weights can be either computed at time n for time $n+1$ and are independent of $z_{1:u}$.

As in Del Moral et al. (2006), we adopted a piecewise linear cooling scheme that had, for 50 time points, $1/\zeta_n$ increase uniformly to 0.05 for the first 10 time points, then uniformly to 0.3 for the next 20 and then uniformly to 1. All other time specifications had the same cooling schedule, in time proportion.

In Figures 1, 2, 3, 4 and Table 2 we can observe our results. Figures 2, 3, 4 and Table 2 provide a comparison of the performance for the two backward kernels. As expected, (18) provides substantial improvements over the reversal kernel (21) with significantly lower weight degeneracy and thus fewer resampling steps. This is manifested in Figure 1 with slighly less dependence (of the samples) for the Gibbs kernel. The CPU times of the two SMC samplers are comparable to MCMC (Table 2 final column) which shows that SMC can markedly improve upon MCMC for similar computational cost (and programming effort).

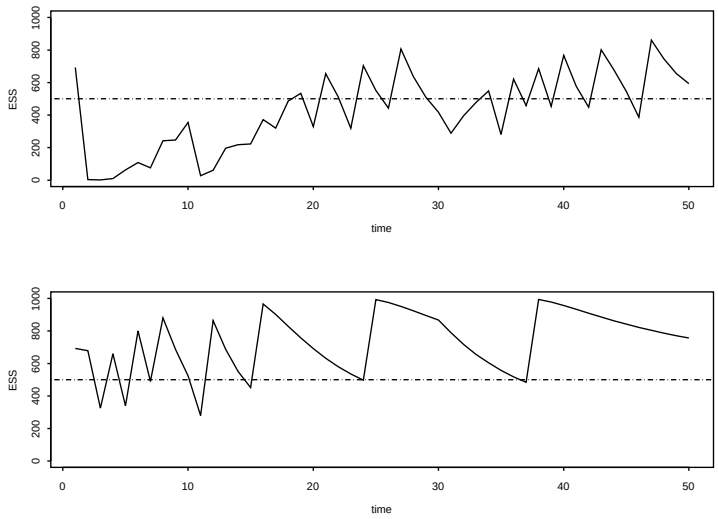

Figure 2: *ESS plots from the binary regression example; 50 time points. The top graph is for reversal kernel (18). We sampled 1000 particles and resampled when the ESS dropped below 500 particles,*

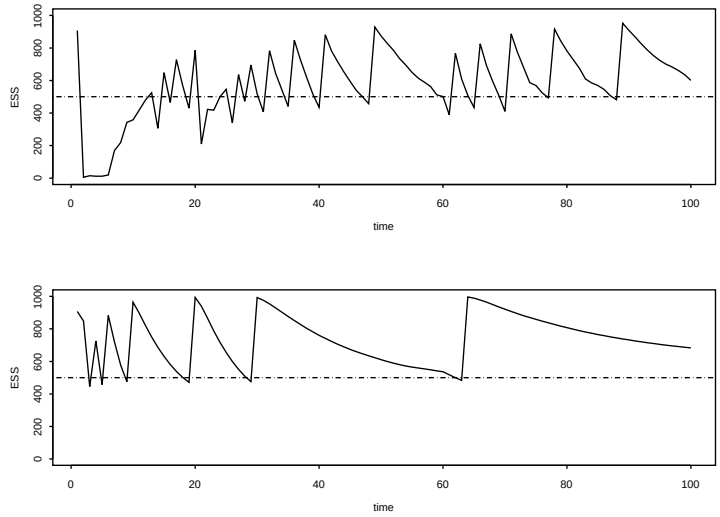

Figure 3: *ESS plots from the binary regression example; 100 time points. The top graph is for reversal kernel (18). We sampled 1000 particles and resampled when the ESS dropped below 500 particles,*

Table 2: *Results from Binary regression example. The first entry is for the reversal (i.e. the first column row entry is the reversal kernel for 50 time points). The CPU time is in seconds.*

Time points	**50**	**100**	**200**
CPU Time	115.33	251.70	681.33
CPU Time	118.93	263.61	677.65
# Times Resampled	29	29	28
# Times Resampled	7	6	8

4.2.5. *Summary*

In this example we have established that SMC samplers are an alternative to MCMC for a binary regression example. This was only at a slight increase in CPU time and programming effort. As a result, we may be able to investigate more challenging problems, especially since we have not utilized all of the SMC strategies (e.g. adaptive methods, in Section 2.2).

We also saw that the adoption of the Gibbs backward kernel (18) provided significant improvements over Resample-Move. This is of interest when the full conditionals are not available, but good approximations of them are. In this case it would be of interest to see if similar results hold, that is, in comparison with the reversal kernel (21). We note that this is not meaningless in the context of artifical distributions, where the rate of resampling may be controlled by ensuring $\pi_{n-1} \approx \pi_n$. This is because we will obtain better performance for the Gibbs kernel for shorter time specifications (and particle number) and hence lower CPU time.

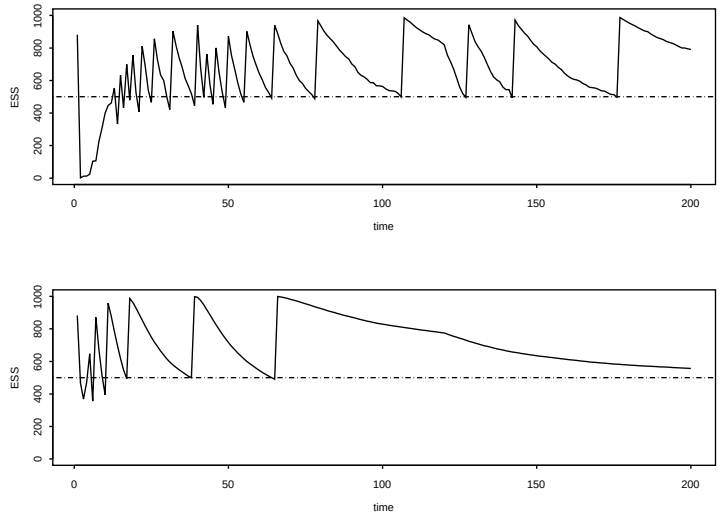

Figure 4: *ESS plots from the binary regression example; 200 time points. The top graph is for reversal kernel (18). We sampled 1000 particles and resampled when the ESS dropped below 500 particles,*

4.3. *Filtering for Partially Observed Processes*

In the following example we consider SMC samplers applied to filtering for partially observed processes. In particular, we extend the approach of Del Moral et al. (2006) for cases with $S_{n-1} \subset S_n$, that is, a sequence of densities with nested supports.

4.3.1. *Model*

We focus upon the Bayesian Ornstein-Uhlenbeck stochastic volatility model (Barndoff-Nielsen and Shepard 2001) found in Roberts et al. (2004). That is, the log-return of an asset X_t at time $t \in [0, T]$ is modelled via the stochastic differential equation (SDE):

$$dX_t = \sigma_t^{1/2} dW_t$$

where $\{W_t\}_{t\in[0,T]}$ is a standard Wiener process. The volatility σ_t is assumed to satisfy the following (Ornstein-Uhlenbeck equation) SDE:

$$d\sigma_t = -\mu\sigma_t dt + dZ_t \tag{30}$$

where $\{Z_t\}_{t\in[0,T]}$ is assumed to be a pure jump Lévy process; see Applebaum (2004) for a nice introduction.

It is well known (Barndoff-Nielsen and Shephard 2001; Applebaum 2004) that for any self-decomposable random variable, there exists a unique Lévy process that satisfies (30); we assume that σ_t has a Gamma marginal, $\mathcal{G}a(\nu, \theta)$. In this case Z_t

is a compound Poisson process:

$$Z_t = \sum_{j=1}^{K_t} \varepsilon_j$$

where K_t is a Poisson process of rate $\nu\mu$ and the ε_j are i.i.d. according to $\mathcal{E}x(\theta)$ (where $\mathcal{E}x$ is the exponential distribution). Denote the jump times of the compound Poisson process as $0 < c_1 < \cdots < c_{k_t} < t$.

Since $X_t \sim \mathcal{N}(0, \sigma_t^*)$, where $\sigma_t^* = \int_0^t \sigma_s ds$ is the integrated volatility, it is easily seen that $Y_{t_i} \sim \mathcal{N}(0, \sigma_i^*)$ with $Y_{t_i} = X_{t_i} - X_{t_{i-1}}$, $0 < t_1 < \cdots < t_u = T$ are regularly spaced observation times and $\sigma_i^* = \sigma_{t_i}^* - \sigma_{t_{i-1}}^*$. Additionally, the integrated volatility is:

$$\sigma_t^* = \frac{1}{\mu}\Big(\sum_{j=1}^{K_t}[1 - \exp\{-\mu(t - c_j)\}]\varepsilon_j - \sigma_0[\exp\{-\mu t\} - 1]\Big)$$

The likelihood at time t is

$$g(y_{t_1:m_t}|\{\sigma_t^*\}) = \prod_{i=1}^{n} \phi(y_{t_i}; \sigma_i^*)\mathbb{I}_{\{t_i<t\}}(t_i)$$

with $\phi(\cdot; a)$ the density of normal distribution of mean zero and variance a and $m_t = \max\{t_i : t_i \leq t\}$. The priors are exactly as Roberts et al. (2004):

$$\begin{array}{ll} \sigma_0|\theta, \nu & \sim \mathcal{G}a(\nu, \theta),\ \nu \sim \mathcal{G}a(\alpha_\nu, \beta_\nu), \\ \mu & \sim \mathcal{G}a(\alpha_\mu, \beta_\mu),\ \theta \sim \mathcal{G}a(\alpha_\theta, \beta_\theta) \end{array}$$

where $\mathcal{G}a(a, b)$ is the Gamma distribution of mean a/b. We take the density, at time t, of the stochastic process, with respect to (the product of) Lebesgue and counting measures:

$$\begin{array}{ll} p_t(c_{1:k_t}, \varepsilon_{1:k_t}, k_t) & = \frac{k_t!}{n^{k_t}}\mathbb{I}_{\{0<c_1<\cdots<c_{k_t}<t\}}(c_{1:k_t})\theta^{k_t}\exp\{-\theta\sum_{j=1}^{k_t}\varepsilon_j\} \times \\ & \frac{(t\mu\nu)^{k_t}}{k_t!}\exp\{-t\mu\nu\}. \end{array}$$

4.3.2. *Simulation Details*

We are thus interested in simulating from a sequence of densities, which at time n (of the sampler) and corresponding $d_n \in (0, T]$ (of the stochastic process) is defined as:

$$\begin{array}{ll} \pi_n(c_{1:k_{d_n}}, \varepsilon_{1:k_{d_n}}, k_{d_n}, \sigma_0, \nu, \mu, \theta|y_{t_1:m_{d_n}}) & \propto g(y_{t_1:m_{d_n}}|\{\sigma_{d_n}^*\})\pi(\sigma_0, \nu, \mu, \theta)\times \\ & p_{d_n}(c_{1:k_{d_n}}, \varepsilon_{1:k_{d_n}}, k_{d_n}). \end{array}$$

As in example 2 of Del Moral et al. (2006) this is a sequence of densities on trans-dimensional, nested spaces. However, the problem is significantly more difficult as the full conditional densities are not available in closed form. To simulate this sequence, we adopted the following technique.

If $k_{d_n} = 0$ we select a birth move which is the same as Roberts et al. (2004). Otherwise, we extend the space by adopting a random walk kernel:

$$q((c_{k_{d_n-1}} - 1, c_{k_{d_n-1}}), c_{k_{d_n}}) \quad \propto \exp\{-\lambda|c_{k_{d_n}} - c_{k_{d_n-1}}|\}\mathbb{I}_{(c_{k_{d_n-1}}-1, n)}(c_{k_{d_n}}).$$

The backward kernel is identical if $c_{k_{d_n}} \in (0, d_{n-1})$ otherwise it is uniform. The incremental weight is then much like a Hastings ratio, but standard manipulations establish that it has finite supremum norm, which means that it has finite variance. However, we found that the ESS could drop, when very informative observations arrive and thus we used the following idea: If the ESS drops, we return to the original particles at time $n-1$ and we perform an SMC sampler which heats up to a very simple (related) density and then make the space extension (much like the tempered transitions method of Neal (1996)). We then use SMC to return to the density we were interested in sampling from.

After this step we perform an MCMC step (the centered algorithm of Roberts et al. (2004)) which leaves π_n invariant allowing with probability 1/2 a Dirac step to reduce the CPU time spent on updating the particles.

4.3.3. *Illustration*

For illustration purposes we simulated $u = 500$ data points from the prior and ran 10000 particles with systematic resampling (threshold 3000 particles). The priors were $\alpha_\nu = 1.0$, $\beta_\nu = 0.5$, $\alpha_\mu = 1.0$, $\beta_\mu = 1.0$, $\alpha_\theta = 1.0$, $\beta_\theta = 0.1$. We defined the target densities at the observation times $1, 2, \ldots, 500$ and set $\lambda = 10$.

If the ESS drops we perform the algorithm with respect to:

$$\pi_n^\zeta(c_{1:k_{d_n}}, \varepsilon_{1:k_{d_n}}, k_{d_n}, \sigma_0, \nu, \mu, \theta | y_{t_1:m_{d_n}}) \propto g(y_{t_1:m_{d_n}} | \{\sigma^*_{d_n}\})^\zeta \pi(\sigma_0, \nu, \mu, \theta) \times p_{d_n}(c_{1:k_{d_n}}, \varepsilon_{1:k_{d_n}}, k_{d_n})$$

for some temperatures $\{\zeta\}$. We used a uniform heating/cooling schedule to $\zeta = 0.005$ and 100 densities and performed this if the ESS dropped to 5% of the particle number.

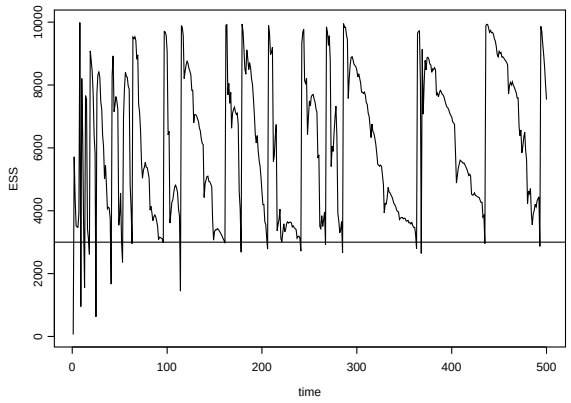

Figure 5: *ESS plot for simulated data from the stochastic volatility example. We ran 10000 particles with resampling threshold (−−) 3000 particles.*

We can see in Figure 5 that we are able to extend the state-space in an efficient manner and then estimate (Figure 6) the filtered and smoothed actual volatility σ_i^*

which, to our knowledge, has not ever been performed for such complex models. It should be noted that we only had to apply the procedure above, for when the ESS drops, 7 times; which illustrates that our original incremental weight does not have extremely high variance. For this example, the MCMC moves can operate upon the entire state-space, which we recommend, unless a faster mixing MCMC sampler is constructed. That is, the computational complexity is dependent upon u (the number of data points). Additionally, due to the required, extra, SMC sampler, this approach is not useful for high frequency data, but is more appropriate for daily returns type data.

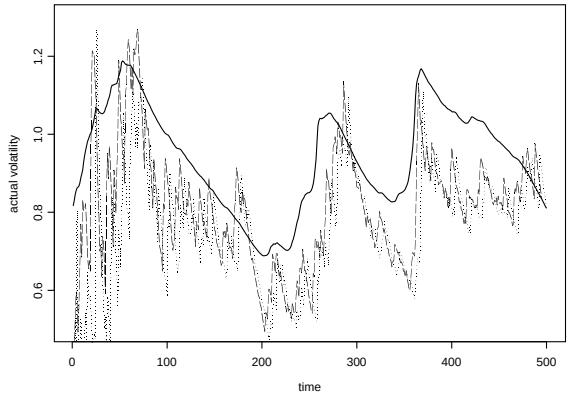

Figure 6: *Actual volatility for simulated data from the stochastic volatility example. We plotted the actual volatility for the final density (full line) filtered (esimated at each timepoint, dot) and smoothed (estimated at each timepoint, lag 5, dash)*

5. CONCLUSION

It is well-known that SMC algorithms can solve, numerically, sequential Bayesian inference problems for nonlinear, non-Gaussian state-space models (Doucet et al. 2001). We have demonstrated (in addition to the work of Chopin, 2002; Del Moral *et al.*, 2006; Gilks and Berzuini, 2001; Neal, 2001) that SMC methods are not limited to this class of applications and can be used to solve, efficiently, a wide variety of problems arising in Bayesian statistics.

We remark that, as for MCMC, SMC methods are not black-boxes and require some expertise to achieve good performance. Nevertheless, contrary to MCMC, as SMC is essentially based upon IS, its validity does not rely on ergodic properties of any Markov chain. Consequently, the type of strategies that may be applied by the user is far richer, that is, time-adaptive proposals and even non-Markov transition kernels can be used without any theoretical difficulties. Such schemes are presented in Jasra et al. (2005a) for trans-dimensional problems.

We also believe that it is fruitful to interpret SMC as a particle approximation of nonlinear MCMC kernels. This provides us with alternative SMC and iterative self-interacting approximation schemes as well as opening the avenue for new nonlinear

algorithms. The key to these procedures is being able to design nonlinear MCMC kernels admitting fixed target distributions; see Andrieu et al. (2006) and Brockwell and Doucet (2006) for such algorithms.

ACKNOWLEDGEMENTS

The second author would like to thank Mark Briers for the simulations of the block sampling example, Adam Johansen for his comments and Gareth W. Peters. The third author would like to thank Chris Holmes for funding and Dave Stephens for discussions on the examples.

REFERENCES

Albert J. H. and Chib S. (1993). Bayesian analysis of binary and polychotomous response data. *J. Amer. Statist. Assoc.* **88**, 669–679.

Andrieu, C., Jasra, A., Doucet, A. and Del Moral, P. (2006). Non-linear Markov chain Monte Carlo via self interacting approximations. *Tech. Rep.*, University of Bristol.

Applebaum D. (2004) *Lévy Processes and Stochastic Calculus*, Cambridge: University Press.

Barndoff Nielsen, O. E. and Shephard, N. (2001). Non-Gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics *J. Roy. Statist. Soc. B* **63** ,167-241, (with discussion). .

Brockwell, A. E. and Doucet, A. (2006). Sequentially interacting Markov chain Monte Carlo for Bayesian computation, *Tech. Rep.*, Carnegie Mellon University, USA.

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

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

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. *Séminaire de Probabilités XXXVII*, (Azéma, J., Emery, M., Ledoux, M. and Yor, M., eds.) *Lecture Notes in Mathematics* **1832,**Berlin: Springer, 415–446.

Del Moral, P., Doucet, A. and Jasra, A. (2006). Sequential Monte Carlo samplers. *J. Roy. Statist. Soc. B* **68**, 411–436.

Del Moral, P. and Miclo, L. (2004). On convergence of chains with occupational self-interactions. *Proc. Roy. Soc. A* **460**, 325–346.

Del Moral, P. and Miclo, L. (2006). Self interacting Markov chains. *Stoch. Analysis Appl.*, **3** 615–660.

Doucet, A., Briers, M. and Sénécal, S. (2006). Efficient block sampling strategies for sequential Monte Carlo. *J. Comp. Graphical Statist.* **15**, 693–711.

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

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

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

Green, P.J. (2003). Trans-dimensional Markov chain Monte Carlo. in *Highly Structured Stochastic Systems*, Oxford: University Press

Holmes, C. C. and Held, L (2006). Bayesian auxiliary variable models for binary and multinomial regression *Bayesian Analysis* **1**, 145–168,

Iba, Y. (2001). Population Monte Carlo algorithms. *Trans. Jap. Soc. Artif. Intell.* **16**, 279–286

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

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

Jasra, A., Stephens, D. A. and Holmes, C.C. (2005b). On population-based simulation for static inference. *Tech. Rep.*, Imperial College London, UK.

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

Neal, R. (1996). Sampling from multimodal distributions via tempered transitions. *Statist. Computing* **6**, 353–366.

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

Roberts, G. O., Papaspiliopoulos, O. and Dellaportas, P. (2004). Bayesian inference for non-Gaussian Ornstein-Uhlenbeck stochastic volatility processes, *J. Roy. Statist. Soc. B* **66**, 369–393.

DISCUSSION

HEDIBERT FREITAS LOPES (*University of Chicago, USA*)

I would like to start by congratulating the authors on this very important contribution to a growing literature on sequential Monte Carlo (SMC) methods. The authors run the extra mile and show how SMC methods, when combined with well established MCMC tools, can be used to entertain not only optimal filtering problems or, as the authors refer to, sequential Bayesian inference problems, but also for posterior inference in general parametric statistical models.

The paper, similarly to several others in this area, builds upon the simple and neat idea behind sampling importance resampling (SIR) algorithms, in which particles from a (sequence of) importance density are appropriately reweighed in order to obtain (sequences of) approximate draws from a target distribution (see, Smith and Gelfand, 1990, and Gordon, Salmond and Smith, 1995, for a few seminal references).

However, unlike in filtering problems where the nature of the state-space evolution suggests updating schemes for the set of particles, there is no natural rule for sequentially, optimally reweighing the particles that would also be practically feasible. Here lies the main contribution of the paper, where the authors introduce the idea of *auxiliary backward Markov kernels*. SMC methods can then be seen as an alternative to MCMC methods when dealing with sequences of probability distributions that avoids, and to a certain extent eliminates, convergence issues, one of the most cumbersome practical problems in the MCMC literature. Additionally, parallelization of the computation comes as a natural by-product of SMC methods.

I would like to hear (or read) what the authors have to say about a few points that I believe will be in the SMC agenda for the next few years: (i) choice of Markov kernels K for (increasingly) high dimensional state vectorsx_n, such as in modern highly structure stochastic systems; (ii) situations where both fixed parameters and state variables are present, how this distinction helps or makes it difficult to propose kernels? (iii) How to accurately compute and use the variance of weights sequentially when resampling, in principle, makes it more difficult to derive general limiting results? (iv) How does the previous issue relate to the problem of sample impoverishment?

In summary, I humbly anticipate that the next few years will witness a great interaction between *MCMCers* and *SMCers*, both theoretically and empirically. Thanks the authors for writing such an interesting paper.

DAVID R. BICKEL (*Pioneer Hi-Bred Intl., Johnston, Iowa, USA*)

The authors added promising innovations in sequential Monte Carlo methodology to the arsenal of the Bayesian community. Most notably, their backward-kernel framework obviates the evaluation of the importance density function, enabling greater flexibility in the choice of algorithms. They also set their work on posterior inference in a more general context by citing results of the observation that particle filters approximate the path integrals studied in theoretical physics (Del Moral 2004).

My first question concerns another recent advance in SMC, the use of the mixture transition kernel

$$\overline{K}_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right) = \sum_{m=1}^{M} \overline{\alpha}_{n,m} \kappa_m\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right),$$

where $\overline{\alpha}_{n,m}$ equals the sum of normalized weights over all particle values that were drawn from the mth mixture component at time $n-1$, and $\kappa_m\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right)$ is an element of the set of M predetermined mixture components (Douc et al. 2007). For example, if the possible transition kernels correspond to Metropolis-Hastings random walk kernels of M different scales chosen by the statistician, then the mixture automatically adapts to the ones most appropriate for the target distribution (Douc et al. 2007). Is there a class of static inference problems for which the backward-kernel approach is better suited, or is it too early to predict which method may perform better in a particular situation?

In his discussion, Hedibert Lopes suggested some opportunities for further SMC research. What areas of mathematical and applied work seem most worthwhile?

I thank the authors for their highly interesting and informative paper.

NICOLAS CHOPIN (*University of Bristol, UK*)

As the authors know already, I am quite enthusiastic about the general SMC framework they developed, and more generally the idea that SMC can outperform MCMC in a variety of 'complex' Bayesian problems. By 'complex', I refer informally to typical difficulties such as: polymodality, large dimensionality (of the parameter, of the observations, or both), strong correlations between components of the considered posterior distribution, etc.

My discussion focuses on the probit example, and particularly the artificial sequence (γ_n) specifically chosen in this application, which I find both intriguing and exciting. As the authors have certainly realised, all the distributions γ_n are equal to the posterior distribution of interest, up to scaling factor ς_n. For a sequence of Gaussian distributions, rescaling with factor ς_n is the same thing as tempering with exponent ς_n^{-2}; therefore, for standard, well-behaved Bayesian problems both approaches can be considered as roughly equivalent. But rescaling is more convenient here for a number of reasons. First, this means that all the Gibbs steps performed within the SMC algorithm are identical to the Gibbs update implemented in the corresponding MCMC algorithm. Thus a clear case is made that, even if update kernels have the same ergodicity properties in both implementations, the SMC implementation provides better estimates (i.e. with smaller variance) than the MCMC one, for the same computational cost.

Second, the fact that all π_n are equivalent, up to appropriate scaling, means that one can combine results from all or part of the iterations, rather than retaining only

the output of the last iteration. I wonder if the authors have some guidance on how this can be done in an optimal way. Third, and more generally, I am excited about the idea of defining a sequence of artificial *models* in order to derive the sequence (γ_n). This should make it easier to derive kernels K_n (typically Gibbs like) that allows for efficient MCMC step within the SMC algorithm. I think this is a very promising line of research. My only concern is that some models involving latent variables may be more difficult to handle, because, in contrast with this probit example, N simulations of the vector of latent variables would need to be carried forward across iterations, which seems memory demanding.

YANAN FAN, DAVID S. LESLIE and MATTHEW P. WAND
(*University of New South Wales, Australia and University of Bristol, UK*)

We would like to congratulate the authors on their efforts in presenting a unified approached to the use of sequential Monte Carlo (SMC) samplers in Bayesian computation. In this, and the companion publication (Del Moral et al. 2006), the authors illustrate the use of SMC as an alternative to Markov chain Monte Carlo (MCMC).

These methods have several advantages over traditional MCMC methods. Firstly, unlike MCMC, SMC methods do not face the sometimes contentious issue of diagnosing convergence of a Markov chain. Secondly, in problems where mixing is chronically slow, this method appear to offer a more efficient alternative, see Sisson *et al* (2006) for example. Finally, as the authors point out, adaptive proposals for transition kernels can easily be applied since the validity of SMC does not rely on ergodic properties of any Markov chain. This last property may give the SMC approach more scope for improving algorithm efficiency than MCMC.

In reference to the binary probit regression model presented in Section 4.2, the authors chose to use a multivariate normal distribution as the initial importance distribution, with parameter value given by simulated estimates from an MCMC sampler. An alternative, more efficient strategy may be to estimate the parameters of the multivariate normal distribution by fitting the frequentist binary probit regression model. One can obtain maximum likelihood and the associated variance-covariance estimates for this, and many other, models using standard statistical software packages. We are currently designing a SMC method to fit a general design generalized linear mixed model (GLMM) (Zhao et al. 2006) and find this approach to work well.

The authors adopt an MCMC kernel, and update the coefficients of the covariates β from its full conditional distribution. If one cannot sample directly from the full conditional distributions, a Metropolis-Hastings kernel may be used. The choice of scaling parameters in such kernels can greatly influence the performance of the sampler, and it is not clear if optimal scaling techniques developed in the MCMC literature are immediately applicable here. In our current work to use these techniques for GLMMs we have found that using the variance-covariance estimate from the frequentist model as a guide for scaling the MH proposal variance works well. In general, can the authors offer any guidance on the properties of optimal MCMC kernels for use with SMC samplers?

PAUL FEARNHEAD (*Lancaster University, UK*)

I would like to congratulate the authors on describing an exciting development in Sequential Monte Carlo (SMC) methods: extending both the range of applications

and the flexibility of the algorithm. It will be interesting to see how popular and useful these ideas will be in years to come.

The ideas in the paper introduce more choice into the design of SMC algorithms, and it is thus necessary to gain practical insights into how to design efficient algorithms. I would thus like to ask for some further details on the examples.

Firstly I would have liked to see more details about the bearings-only tracking example in Section 4.1. For example, how informative were the priors used, and what did a typical path of the target look like? The choice of prior can have a substantial impact on the efficiency of the standard SMC algorithms that you compare with, with them being very inefficient for relatively diffuse priors (there can be simple ways round this by sampling your first set of particles conditional on the first or first two observations). While the efficiency of EKF approximations to the model can depend on the amount of non-linearity in the observation equation, which in turn depends on the path of the target (with the non-linearity being extreme if the target passes close to the observer). The EKF has a reputation for being unstable on the bearings-only tracking problem (e.g. Gordon *et al.* 1993), so it is somewhat suprising that this block sampler works so well on this problem. (Perhaps the EKF approximation works well because of the informative initial conditions - i.e. a known position and velocity for each particle?)

Also, there are better ways of implementing a SMC algorithm than those that you compare to. In particular, the proposal distribution can be chosen to take into account the information in the most recent observation (Carpenter *et al.* 1999) - which is of particular importance here as the observations are very accurate. How much more efficient is using information from 10 observations as compared to this simpler scheme which uses information from a single observation?

Finally, a comment on your last example (Section 4.3). This is a very challenging problem, and your results are very encouraging, but I have a slight concern about your method whereby if the ESS of the particles drops significantly at an iteration you go back and re-propose the particles from a different proposal. The text reads as though you throw away the particles you initially proposed. If so, does this not introduce a bias into the algorithm (as you will have fewer particles in areas where their importance sampling weight will be large - as these particles will tend to get discarded at such a step)? For batch problems, a simple (and closely related) alternative would be to just let the amount of CPU time/number of particles generated vary with iteration of your algorithm - spending more effort on iterations where the ESS would otherwise drop significantly, and less on iterations where ESS is more robust. For example, this could be done by generating enough particles until the ESS rises above some threshold.

STEVEN L. SCOTT (*University of Southern California, USA*)

Congratulations to Del Moral, Doucet, and Jasra for an informative paper summarizing the current state of the art in sequential Monte Carlo (SMC) methods. SMC has become the tool of choice for Bayesian analysis of nonlinear dynamic models with continuous state spaces. Particularly welcome in the paper are ideas drawn from the MCMC literature. Auxiliary variables and kernel based proposals broaden the SMC toolkit for an audience already comfortable with MCMC.

One goal of the article is to demonstrate SMC as an alternative to MCMC for non-dynamic problems. I wish to make two points regarding the probit regression example from Section 4.2, the only example in Section 4 that is a non-dynamic problem. The first is a clarification about the poor performance of Albert and Chib's

method, which would not be so widely used if its typical performance were as bad as suggested. When Albert and Chib's method mixes slowly, its poor performance can usually be attributed to a "nearly-separating" hyperplane, where most observations have success probabilities very close to 0 or 1. The authors' simulation appears to involve such a phenomenon. If $x_{20} \approx x_2 \approx 1.0$ then β_{20} adds about 30 units to the linear predictor, while β_2 subtracts about 15, leaving the latent probit about 15 standard deviations from zero. There is no information in the paper about the values of the covariates, but a value of 1 is reasonable if the covariates were simulated from standard Gaussian or uniform distributions.

The second point is that the comparison between the authors' SMC algorithm and Albert and Chib is not quite fair because SMC uses a "trick" withheld from its competitor. SMC achieves its efficiency by manipulating the variance of the latent data. Several authors have used this device in a computationally trivial improvement to Albert and Chib's method. The improvement assumes the latent variables have variance ζ, where Albert and Chib assume $\zeta = 1$ for identifiability. To introduce this improvement in the MCMC algorithm one need only modify the variance of the truncated normal distribution for z and replace the draw from $p(\beta|z)$ with a draw from $p(\beta, \zeta|z)$. Thus computing times for the improved and standard algorithms are virturally identical. Identifiability is restored during post processing by dividing each β by the $\sqrt{\zeta}$ from the same Gibbs iteration. Mathematical results due to Meng and van Dyk (1999), Liu and Wu (1999), and Lavine (2003) guarantee that the post processed β's have the desired stationary distribution and mix at least as rapidly as those from the identified model. van Dyk and Meng (2001) illustrate the effects of increased mixing on a variety of examples from the canon of models amenable to data augmentation, including probit regression. Liu and Wu (1999) compare the performance of the improved and standard Albert and Chib algorithms and find substantial mixing improvements in the presence of nearly separating hyperplanes similar to the simulation described by the current paper. The authors carefully acknowledge other Monte Carlo samplers for probit regression but limit their comparison to Albert and Chib (1993) because of its prevalence in applied work. However, given the nature of this particular SMC algorithm and the minimal burden imposed by the method described above, it would seem more appropriate to compare SMC to "improved" rather than "standard" Albert and Chib.

REPLY TO THE DISCUSSION

Firstly we thank the discussants for a set of both stimulating and useful comments. We hope that our work and the comments of the discussants will encourage future research in this developing field of research.

Convergence Diagnosis

As noted in the paper and outlined by Lopes and Fan, Leslie and Wand, SMC methods do not rely upon the ergodicity properties of any Markov kernel and as such do not require convergence checks as for MCMC. However, we feel that we should point out that the performance of the algorithm needs to be monitored closely. For example, the ESS needs to be tracked (to ensure the algorithm does not 'crash' to a single particle) and it is advisable to observe the sampled parameters, running the algorithm a few times to check consistent answers; see also Chopin (2004) for more advice.

Optimal MCMC kernels

Fan, Leslie and Wand and Lopes ask about constructing optimal MCMC kernels. We discuss potential strategies that may add to the ideas of the discussants. We clarify that, on observation of the expression of the asymptotic variance in the central limit theorem (Del Moral et al. (2006), Proposition 2), it can be deduced that the faster the kernel mixes the smaller the variance, the better algorithm (in this sense). One way to construct optimal MCMC kernels, i.e. close to iid sampling from π_n, might be to adopt the following adaptive strategy. We will assume that $\pi_{n-1} \approx \pi_n$, (which can be achieved, by construction, in problems where MCMC will typically be used) thus we can seek to use the particles at time $n-1$ to adapt an MCMC kernel at time n. Two strategies that might be used are:

(i) To adopt Robbins Monro type procedures so that the kernel is optimal in some sense. For example, via the optimal scaling of Roberts et al. (1997) for random walk Metropolis. That is, optimal scaling can be useful as it can attempt to improve the efficiency of the MCMC kernel.

(ii) Attempting to approximate the posterior at time n using the particles at time $n-1$; e.g. by using a mixture approximation in the MH independence sampler, as proposed by Andrieu and Moulines (2006) for adaptive MCMC.

To reply to Lopes' point (ii), the problem he mentions is quite difficult. In the context of state-space models, the introduction of MCMC steps of fixed computational complexity at each time step is not sufficient to reduce the accumulation of errors (i.e. does not make the dynamic model ergodic). Two algorithms combining SMC and stochastic approximation procedures have been proposed to solve this problem; see Andrieu, Doucet and Tadić (2005) and Poyadjis, Doucet and Singh (2005).

The Backward Kernel

In response to Bickel's comment on the use of the $D-$kernel procedure of Douc et al. (2006) against the backward kernel procedure. In the context that the discussant mentions, it would not be possible (except for toy problems) to use the $D-$kernel procedure; it is typically impossible to compute pointwise a MH kernel and thus the importance weights.

Variance of the weights

Lopes, in point (iii) and (iv) asks about the calculation of the variance of the weights. Typically the variance of particle algorithms is estimated (even in the presence of resampling) via the coefficient of variation (e.g. Liu (2001)) of the unnormalized weight (suppressing the time index):

$$C_v = \frac{\sum_{i=1}^N (W^{(i)} - \frac{1}{N}\sum_{j=1}^N W^{(j)})^2}{(N-1)[\frac{1}{N}\sum_{i=1}^N W^{(i)}]^2}.$$

This is related, to the easier to interpret ESS, estimated as:

$$\text{ESS} = \left(\sum_{i=1}^N \left[\frac{W^{(i)}}{\sum_{j=1}^N W^{(j)}} \right]^2 \right)^{-1}$$

which is a proxy for the number of independent samples. These are simply indicators of the variance, and the latter quantity is used as a criterion to judge the degeneracy of the algorithm; that is, a low ESS suggests that the algorithm does not contain many diverse samples and that resampling should be performed. Note, the ESS can be misleading; see the discussion in Chopin (2002). In theory, the resampling step can make the analysis of the algorithm more challenging. The mathematical techniques are based upon measure-valued processes; see Del Moral (2004) and the references therein.

The Tracking Example

In response to Fearnhead's comment about our prior. We selected a reasonably informative prior; i.e. a Gaussian of mean equal to the true initial values and covariance equal to the identity matrix. However, we believe that performance would not degrade drastically if a more diffuse (but not very vague) prior was used. Indeed, in the block sampling strategy described in Section 4.1, the particles at the time origin would be eventually sampled according to an approximation of $p(x_1|y_{1:R})$ at time R. In the scenarios considered here, the EKF diverged for a small percentage of the simulated paths of length 100. However, this divergence never occurred on a path of length $R \leq 10$ which partly explains why it was possible to use it to build efficient importance sampling densities. In scenarios with an extreme nonlinearity, we do agree with Fearnhead that it is unlikely that such an importance sampling distribution would work well. An importance sampling distribution based on the Unscented Kalman filter might prove more robust.

The Probit Example

We begin firstly, by stating the objectives of this example explicitly and then to address the points of the discussants one-by-one. The intention of presenting the simulations were two-fold:

- To illustrate, in a static setting, that even when MCMC kernels fail to mix quickly, that the combination of:

  (i) A large population of samples.

  (ii) Tempered densities.

  (iii) Interaction of the particles.

  can significantly improve upon MCMC for similar coding effort and CPU time.

- That the usage of the backward kernel (17) can substantially improve upon the reversal kernel (20) in terms of variance of the importance weights.

In response to Chopin's point on scaling. It seems that, given the scale parameters, using the idea of reweighting the tempered densities could be used, in order to use the samples targeting those densities other than that of interest. However, in this case, it is clear that this is not a sensible strategy for those densities far away, in some sense (e.g. high variance models, as will be required to induce the tempering effect in the algorithm), as importance weights will have high variance. In the case that the scale parameters are to be determined, the problem of obtaining optimal parameters (in terms of minimizing the variance of importance weights, functionals

of interest etc) is very difficult as noted in Section 6 of Del Moral et al. (2006); some practical strategies are outlined there.

Chopin's point on artificial models is quite insightful and he has identified a particular extension of the sequence of densities idea. We note that such procedures have appeared previously, for example in (Hodgson, 1999). Such references may provide further ideas in developing artificial models in different contexts and hence new sequences of densities to improve the exploration ability of the sampler.

We hope that we have clarified, with our second point, to Fan, Leslie and Wand, that indeed MH kernels could be used, but we were more interested in exploring a different part of the methodology. As Fan, Leslie and Wand note, the initial importance distribution could be improved, and they suggest a better approach than we adopted; however, from a practical point of view it can be much easier to implement our strategy (that is, the code has already been written for the MCMC steps).

In response to Scott's thorough comments on our probit example. The first point, which is accurate is not so relevant, in terms of what we set out to achieve with this example. One objective of SMC samplers, in static contexts, is somehow to try to remove some of the difficulties of considering how and why MCMC kernels do not mix. In essence, for many statistical problems (e.g. stochastic volatility modelling (Kim et al. 1998; Roberts et al. 2004)) it can be very difficult to design specific MCMC samplers, such that we have identified the difficulties of the 'vanilla' sampler and then dealt with them. As a result, SMC methods are an attempt to produce *generic* methodology, which can improve over standard MCMC methods without needing too much target specific design (although clearly, this can improve the simulations), but also more freedom in sampler design. We hope that this was demonstrated in our example.

The second point of Scott on the comparison with MCMC is not quite accurate. As noted above, but perhaps not clearly enough in the paper, we wanted to demonstrate that slowly mixing kernels can be improved upon using sequences of densities, resampling and a population of samples. As a result, the 'trick' comment does not take into account that the population and resampling steps allow the algorithm to consider a vast amount of information simultaneously to improve the exploration of the target. Whilst, if we wanted a realistic comparison we could have used a superior MCMC method; however, there can be examples where more advanced MCMC techniques do not work well (see for instance Neal (1996) for examples with both tempered transitions and simulated tempering). More simply put, if the algorithm outlined by Scott mixes poorly, the intention is to use SMC to improve upon it. In Scott's example, we might achieve this via using sequences of densities with pseudo prior distributions $\{p_n(\zeta)\}$ converging close to Dirac measure on the set $\{1\}$.

The Stochastic Volatility Example

We respond to the comment of Fearnhead concerning the bias of our scheme involving re-proposing the particles. We begin by clarifying exactly what we do:

- If the ESS drops below 5% of the particle number, after a transition at time n, we return to the particles (and weights) before the transition.
- We change (increase) the sequence of densities so that we perform SMC samplers on increasingly more simple densities. Then we extend the state-space and use SMC samplers to return the particles to the density of which we were interested in sampling.

In view of the above comments, the bias of this procedure can be thought of as similar to ordinary, dynamic resampling SMC techniques. That is, resampling upon the basis of the ESS. Whilst we do not have a theoretical justification for this method (however, we anticipate that we can use the methods of Douc and Moulines (2006) as in dynamic resampling) it provides a simple way to deal with the problem of consecutive densities with regions of high probability in different parts of the state-space - for a fixed $O(N)$ complexity. It is not clear then, that we would have fewer particles where the importance weight is large: firstly, the weight is now different from the first scheme, secondly we would expect that our particle approximation of π_n to be fairly accurate, given the tempering procedure adopted; there is no restriction upon the regions of the space that the particles are allowed to visit.

In response to the suggested idea of adding particles to ensure that the ESS does not drop too far. The first difficulty is when the consecutive densities are so different that it may take a large number of particles (e.g. $10N$) to ensure that our algorithm does not degenerate; this may not be feasible in complex problems due to storage costs. The second drawback that we feel that this procedure has, is that it is not explicitly trying to solve the problem at hand; in effect it is a brute force approach which may not work for feasible computational costs.

ADDITIONAL REFERENCES IN THE DISCUSSION

Andrieu, C., Doucet, A. and Tadić, V. (2005). Online simulation-based methods for parameter estimation in nonlinear non-Gaussian state-space models, *Proc. Control and Decision Conference*.

Andrieu, C. and Moulines, E. (2006). On the ergodicity properties of some adaptive MCMC algorithms, *Ann. Appl. Prob.* **16**, 1462–1505.

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

Douc, R. and Moulines, E. (2006). Limit theorems for weighted samples with applications to sequential Monte Carlo Methods. *Tech. Rep.*, Ecole Polytechnique Palaiseau, France.

Douc, R., Guillin, A., Marin, J. M. and Robert, C. P. (2007). Convergence of adaptive sampling schemes. *Ann. Statist.* (in press).

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

Hodgson, M. E. A. (1999). A Bayesian restoration of an ion channel. *J. Roy. Statist. Soc. B* **61**, 95–114.

Kim, S., Shephard, N. and Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models, *Rev. Econ. Studies* **65**, 361–393.

Lavine, M. (2003). A Marginal ergodic theorem. *Bayesian Statistics 7* (J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West, eds.) Oxford: University Press, 577–586.

Liu, J. S. and Wu, Y.-N. (1999). Parameter expansion for data augmentation. *J. Amer. Statist. Assoc.* **94**, 1264–1274.

Meng, X.-L. and van Dyk, D. A. (1999). Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika* **86**, 301–320

Poyadjis, G., Doucet, A. and Singh, S. S. (2005). Maximum likelihood parameter estimation using particle methods, *Proc. Joint Statistical Meeting, USA*.

Roberts, G. O., Gelman, A. and Gilks, W. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms, *Ann. Appl. Prob.* **7**, 110–120.

Sisson, S. A., Fan, Y. and Tanaka, M. M. (2006). Sequential Monte Carlo without likelihoods. *Tech. Rep.*, University of New South Wales, Australia.

van Dyk, D. A. and Meng, X.-L. (2001). The art of data augmentation. *J. Comp. Graphical Statist.* **10**, 1–111 (with discussion).

Zhao, Y., Staudenmayer, J., Coull, B. A. and Wand, M. P. (2006). General design Bayesian generalized linear mixed models. *Statist. Science* **21**, 35–51.