Sequential Monte Carlo for Bayesian Computation

Pierre Del Moral Arnaud Doucet
Université Nice Sophia Antipolis, FR *University of British Columbia, CA*

Ajay Jasra
University of Cambridge, 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 & 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 $\pi_n(\mathbf{x}_n)$

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 Associate in the Department of Engineering at the University of Cambridge.

with respect to 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 $\gamma_n : E_n \rightarrow \mathbb{R}^+$ 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 & 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 $g\left(y_{1:m_t} | \{c_n, \varepsilon_n\}_{n \geq 1}\right) = g(y_{1:m_t} | c_{1:k_t}, \varepsilon_{1:k_t})$ 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 $\mathbf{x}_n = \left(c_{1:k_{d_n}}, \varepsilon_{1:k_{d_n}}\right)$ and

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

where $c_0 = 0$ by convention. These target distributions are all defined on the same space $E_n = E = \biguplus_{k=1}^{\infty} A_k \times \mathbb{R}^k$ 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 $\biguplus_{k=1}^{\infty} A_{k,d_n} \times \mathbb{R}^k$ 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 $\varphi : E_n \to \mathbb{R}$

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

Throughout we will denote $\int_{E_n} \varphi\left(\mathbf{x}_n\right) \pi_n\left(\mathbf{x}_n\right) d\mathbf{x}_n$ 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 & 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 & 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 & 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 & Berzuini (2001)); Bayesian probit regression (e.g. Albert & 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

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

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

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 $K_n : E_{n-1} \rightarrow \mathcal{P}(E_n)$ (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

$$\eta_n\left(\mathbf{x}_n\right) = \eta_{n-1} K_n\left(\mathbf{x}_n\right) = \int \eta_{n-1}\left(d\mathbf{x}_{n-1}\right) K_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right). \tag{4}$$

2.2. *Selection of Transition Kernels*

It is clear that the optimal importance distribution, in the sense of minimizing the variance of (3), is $\eta_n\left(\mathbf{x}_n\right) = \pi_n\left(\mathbf{x}_n\right)$. Therefore, the optimal transition kernel is simply $K_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right) = \pi_n\left(\mathbf{x}_n\right)$. 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 $\mathbf{x}_{n-1} = \left(\mathbf{u}_{n-1}, \mathbf{v}_{n-1}\right) \in E_{n-1} = U_{n-1} \times V_{n-1}$ to $\mathbf{x}_n = \left(\mathbf{u}_{n-1}, \mathbf{v}_n\right) \in E_n = U_{n-1} \times V_n$ ($V_n \neq \emptyset$). We adopt the following kernel

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

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

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

In order to select $q_n\left(\mathbf{x}_{n-1}, \mathbf{v}_n\right)$, 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

$$q_n^{\text{opt}}\left(\mathbf{x}_{n-1}, \mathbf{v}_n\right) = \pi_n\left(\left.\mathbf{v}_n\right| \mathbf{u}_{n-1}\right) \tag{5}$$

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

$$w_n\left(\mathbf{x}_n\right) = \frac{\gamma_n\left(\mathbf{u}_{n-1}\right)}{\eta_{n-1}\left(\mathbf{u}_{n-1}\right)}. \tag{6}$$

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

In cases where it is possible to sample from $\pi_n\left(\left.\mathbf{v}_n\right| \mathbf{u}_{n-1}\right)$ but impossible to compute $\gamma_n\left(\mathbf{u}_{n-1}\right)$ and/or $\eta_{n-1}\left(\mathbf{u}_{n-1}\right)$, we can use an attractive property of IS: we do not need to compute exactly (6), we can use an unbiased estimate of it. We have the identity

$$\gamma_n\left(\mathbf{u}_{n-1}\right)=\widehat{\gamma}_n\left(\mathbf{u}_{n-1}\right)\int\frac{\gamma_n\left(\mathbf{u}_{n-1},\mathbf{v}_n\right)}{\widehat{\gamma}_n\left(\mathbf{u}_{n-1},\mathbf{v}_n\right)}\widehat{\pi}_n\left(\left.\mathbf{v}_n\right|\mathbf{u}_{n-1}\right)d\mathbf{v}_n \tag{7}$$

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

$$\frac{1}{\eta_{n-1}\left(\mathbf{u}_{n-1}\right)}=\frac{1}{\widehat{\eta}_{n-1}\left(\mathbf{u}_{n-1}\right)}\int\frac{\widehat{\eta}_{n-1}\left(\mathbf{u}_{n-1},\mathbf{v}_{n-1}\right)}{\eta_{n-1}\left(\mathbf{u}_{n-1},\mathbf{v}_{n-1}\right)}\eta_{n-1}\left(\left.\mathbf{v}_{n-1}\right|\mathbf{u}_{n-1}\right)d\mathbf{v}_{n-1} \tag{8}$$

where $\widehat{\eta}_{n-1}\left(\mathbf{u}_{n-1},\mathbf{v}_{n-1}\right)$ is selected as an approximation of $\eta_{n-1}\left(\mathbf{u}_{n-1},\mathbf{v}_{n-1}\right)$ such that $\int\widehat{\eta}_{n-1}\left(\mathbf{u}_{n-1},\mathbf{v}_{n-1}\right)d\mathbf{v}_{n-1}$ can be computed analytically. So if we can sample from $\eta_{n-1}\left(\left.\mathbf{v}_{n-1}\right|\mathbf{u}_{n-1}\right)$, we can calculate an unbiased estimate of (8). This idea has a limited range of applications as in complex cases we do not necessarily have a closed-form expression for $\eta_{n-1}\left(\mathbf{x}_{n-1}\right)$. However, if one has resampled particles at time $k\leq n-1$, then one has (approximately) $\eta_{n-1}\left(\mathbf{x}_{n-1}\right)=\pi_k K_{k+1}K_{k+2}\cdots K_{n-1}\left(\mathbf{x}_{n-1}\right)$.

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 $\pi_n\left(\left.\mathbf{v}_n\right|\mathbf{u}_{n-1}\right)$ 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 $\gamma_n\left(\mathbf{u}_{n-1}\right)$ and we need an approximation $\widehat{\pi}_n\left(\left.\mathbf{v}_n\right|\mathbf{u}_{n-1}\right)$ of $\pi_n\left(\left.\mathbf{v}_n\right|\mathbf{u}_{n-1}\right)$ to compute an unbiased estimate of it with low variance. Alternatively, we can simply use the following transition kernel

$$K_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)=\mathbb{I}_{\mathbf{u}_{n-1}}\left(\mathbf{u}_n\right)\widehat{\pi}_n\left(\left.\mathbf{v}_n\right|\mathbf{u}_{n-1}\right) \tag{9}$$

and the associated importance weight is given by

$$w_n\left(\mathbf{x}_n\right)=\frac{\gamma_n\left(\mathbf{u}_{n-1},\mathbf{v}_n\right)}{\eta_{n-1}\left(\mathbf{u}_{n-1}\right)\widehat{\pi}_n\left(\left.\mathbf{v}_n\right|\mathbf{u}_{n-1}\right)}. \tag{10}$$

Proceeding this way, we bypass the estimation of $\gamma_n\left(\mathbf{u}_{n-1}\right)$ which appeared in (6). However, we still need to compute $\eta_{n-1}\left(\mathbf{u}_{n-1}\right)$ 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 (10) as

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

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 $K_{n,\widehat{\pi}_{n-1}^N}(\mathbf{x}_{n-1}, \mathbf{x}_n)$. This was proposed and justified theoretically in Crisan & 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 (4) 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

$$K_n(\mathbf{x}_{n-1}, \mathbf{x}_n) = \sum_{m=1}^{M} \alpha_{n,m}(\mathbf{x}_{n-1}) K_{n,m}(\mathbf{x}_{n-1}, \mathbf{x}_n) \tag{12}$$

where $\alpha_{n,m}(\mathbf{x}_{n-1}) \geq 0$, $\sum_{m=1}^{M} \alpha_{n,m}(\mathbf{x}_{n-1}) = 1$ and $\{K_{n,m}\}$ is a collection of transition kernels. Unfortunately, the direct calculation of the importance weight (4) associated to (12) 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, (12) 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

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

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

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

The associated importance weight satisfies

$$\begin{aligned} w_n(\mathbf{x}_{1:n}) &= \frac{\widetilde{\gamma}_n(\mathbf{x}_{1:n})}{\eta_n(\mathbf{x}_{1:n})} \\ &= w_{n-1}(\mathbf{x}_{1:n-1})\, \widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}_n). \end{aligned} \tag{13}$$

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 ($N \rightarrow \infty$) 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 kernel L_{n-1}.

2.3.1. *Optimal backward kernels*

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

$$L_k^{\text{opt}}\left(\mathbf{x}_{k+1},\mathbf{x}_k\right)=\frac{\eta_k\left(\mathbf{x}_k\right)K_{k+1}\left(\mathbf{x}_k,\mathbf{x}_{k+1}\right)}{\eta_{k+1}\left(\mathbf{x}_{k+1}\right)} \tag{14}$$

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

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

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 (14). The key point is that, even if they are different from from (14), 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 $\eta_n\left(\mathbf{x}_n\right)$ (i.e. not using $L_k^{\text{opt}}\left(\mathbf{x}_{k+1},\mathbf{x}_k\right)$) is that the variance of the Monte Carlo estimates based upon (13) will be larger. For example, even if we set $\pi_n\left(\mathbf{x}_n\right)=\pi\left(\mathbf{x}_n\right)$ and $K_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)=K\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)$ is an ergodic MCMC kernel of invariant distribution π then the variance of $w_n\left(\mathbf{x}_{1:n}\right)$ will fail to stabilize (or become infinite in some cases) over time for any backward kernel $L_k\left(\mathbf{x}_{k+1},\mathbf{x}_k\right)\neq L_k^{\text{opt}}\left(\mathbf{x}_{k+1},\mathbf{x}_k\right)$ 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 $\left\{L_k\left(\mathbf{x}_{k+1},\mathbf{x}_k\right)\right\}$ for $k=1,...,n-2$ have already been selected and we are interested in some approximations of $L_{n-1}^{\text{opt}}\left(\mathbf{x}_n,\mathbf{x}_{n-1}\right)$ controlling the evolution of the variance of $w_n\left(\mathbf{x}_{1:n}\right)$.

2.3.2. *Suboptimal backward kernels*

- *Substituting* π_{n-1} *for* η_{n-1}. Equation (14) suggests that a sensible suboptimal strategy consists of substituting π_{n-1} for η_{n-1} to obtain

$$L_{n-1}\left(\mathbf{x}_n,\mathbf{x}_{n-1}\right)=\frac{\pi_{n-1}\left(\mathbf{x}_{n-1}\right)K_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)}{\pi_{n-1}K_n\left(\mathbf{x}_n\right)} \tag{15}$$

which yields

$$\widetilde{w}_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)=\frac{\gamma_n\left(\mathbf{x}_n\right)}{\int\gamma_{n-1}\left(d\mathbf{x}_{n-1}\right)K_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)}. \tag{16}$$

It is often more convenient to use (16) than (14) 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 (14) is equal to (15).

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

$$K_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)=\mathbb{I}_{\mathbf{u}_{n-1}}\left(\mathbf{u}_n\right)\pi_n\left(\mathbf{v}_n\middle|\,\mathbf{u}_{n-1}\right) \tag{17}$$

In this case (15) and (16) are given by

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

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

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

$$L_{n-1}\left(\mathbf{x}_n,\mathbf{x}_{n-1}\right)=\mathbb{I}_{\mathbf{u}_n}\left(\mathbf{u}_{n-1}\right)\widehat{\pi}_{n-1}\left(\mathbf{v}_{n-1}\middle|\,\mathbf{u}_{n-1}\right), \tag{18}$$

$$\widetilde{w}_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)=\frac{\gamma_n\left(\mathbf{u}_{n-1},\mathbf{v}_n\right)\widehat{\pi}_{n-1}\left(\mathbf{v}_{n-1}\middle|\,\mathbf{u}_{n-1}\right)}{\gamma_{n-1}\left(\mathbf{u}_{n-1},\mathbf{v}_{n-1}\right)\widehat{\pi}_n\left(\mathbf{v}_n\middle|\,\mathbf{u}_{n-1}\right)}. \tag{19}$$

• *MCMC Kernels.* A generic alternative approximation of (15) 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 & Berzuini (2001). It is given by

$$L_{n-1}\left(\mathbf{x}_n,\mathbf{x}_{n-1}\right)=\frac{\pi_n\left(\mathbf{x}_{n-1}\right)K_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)}{\pi_n\left(\mathbf{x}_n\right)} \tag{20}$$

and will be a good approximation of (15) if $\pi_{n-1}\approx\pi_n$; note that (20) is the reversal Markov kernel associated with K_n. In this case, the incremental weight satisfies

$$\widetilde{w}_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)=\frac{\gamma_n\left(\mathbf{x}_{n-1}\right)}{\gamma_{n-1}\left(\mathbf{x}_{n-1}\right)}. \tag{21}$$

This expression (21) is remarkable as it is easy to compute and valid *irrespective* of the MCMC kernel adopted. It is also counter-intuitive: if $K_n\left(\mathbf{x}_{n-1},\mathbf{x}_n\right)$ is mixing quickly so that $\mathbf{X}_n^{(i)}\sim\pi_n$ 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 (15), 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

$$L_{n-1}\left(\mathbf{x}_n, \mathbf{x}_{n-1}\right) = \frac{\pi_n\left(\mathbf{x}_{n-1}\right) K_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right)}{\int_{S_{n-1}} \pi_n\left(\mathbf{x}_{n-1}\right) K_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right) d\mathbf{x}_{n-1}} \tag{22}$$

but the denominator of this expression is different from $\pi_n\left(\mathbf{x}_n\right)$ 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 (12), one should select $L_{n-1}\left(\mathbf{x}_n, \mathbf{x}_{n-1}\right)$ as a mixture

$$L_{n-1}\left(\mathbf{x}_n, \mathbf{x}_{n-1}\right) = \sum_{m=1}^{M} \beta_{n-1,m}\left(\mathbf{x}_n\right) L_{n-1,m}\left(\mathbf{x}_n, \mathbf{x}_{n-1}\right) \tag{23}$$

where $\beta_{n-1,m}\left(\mathbf{x}_n\right) \geq 0$, $\sum_{m=1}^{M} \beta_{n-1,m}\left(\mathbf{x}_n\right) = 1$ and $\{L_{n-1,m}\}$ is a collection of backward transition kernels. Using (14), it is indeed easy to show that the optimal backward kernel corresponds to

$$\beta_{n-1,m}^{\text{opt}}\left(\mathbf{x}_n\right) \propto \int \alpha_{n,m}\left(\mathbf{x}_{n-1}\right) \eta_{n-1}\left(\mathbf{x}_{n-1}\right) K_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right) d\mathbf{x}_{n-1},$$

$$L_{n-1,m}^{\text{opt}}\left(\mathbf{x}_n, \mathbf{x}_{n-1}\right) = \frac{\alpha_{n,m}\left(\mathbf{x}_{n-1}\right) \eta_{n-1}\left(\mathbf{x}_{n-1}\right) K_n\left(\mathbf{x}_{n-1}, \mathbf{x}_{nn}\right)}{\int \alpha_{n,m}\left(\mathbf{x}_{n-1}\right) \eta_{n-1}\left(\mathbf{x}_{n-1}\right) K_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right) d\mathbf{x}_{n-1}}.$$

Various approximations to $\beta_{n-1,m}^{\text{opt}}\left(\mathbf{x}_n\right)$ and $L_{n-1,m}^{\text{opt}}\left(\mathbf{x}_n, \mathbf{x}_{n-1}\right)$ 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 $\mathbb{P}\left(M_n = m\right) = \alpha_{n,m}\left(\mathbf{x}_{n-1}\right)$ and perform IS on the extended space. This yields an incremental importance weight equal to

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

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 $X_1^{(i)} \sim \eta_1$ and compute $W_1^{(i)} \propto w_1\left(X_1^{(i)}\right)$. If ESS<Threshold, resample the particle representation $\left\{W_1^{(i)}, X_1^{(i)}\right\}$.

- **At time** n; $n \geq 2$. Sample $X_n^{(i)} \sim K_n\left(X_{n-1}^{(i)}, \cdot\right)$ and compute $W_n^{(i)} \propto W_{n-1}^{(i)} \widetilde{w}_n\left(X_{n-1}^{(i)}, X_n^{(i)}\right)$. If ESS<Threshold, resample the particle representation $\left\{W_n^{(i)}, X_n^{(i)}\right\}$.

The target π_n is approximated through

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

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

$$\frac{Z_n}{Z_{n-1}} = \frac{\int \gamma_n\left(\mathbf{x}_n\right) d\mathbf{x}_n}{\int \gamma_{n-1}\left(\mathbf{x}_{n-1}\right) d\mathbf{x}_{n-1}} \text{ by } \widehat{\frac{Z_n}{Z_{n-1}}} = \sum_{i=1}^{N} W_{n-1}^{(i)} \widetilde{w}_n\left(\mathbf{X}_{n-1}^{(i)}, \mathbf{X}_n^{(i)}\right). \quad (24)$$

Alternatively, it is possible to use path sampling (Gelman & 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 $\mathbf{X}_n \sim Q_n\left(\mathbf{X}_{n-1}, \cdot\right)$ and the distribution μ_n of $\mathbf{X}_n$ satisfies $\mu_n = \mu_{n-1} Q_n$. Nonlinear Markov chains are such that $\mathbf{X}_n \sim Q_{\mu_{n-1},n}\left(\mathbf{X}_{n-1}, \cdot\right)$, 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

$$\mu_n = \mu_{n-1} Q_{n,\mu_{n-1}}. \quad (25)$$

In a similar fashion to MCMC, it is possible to design nonlinear Markov chain kernels admitting a fixed target π (Del Moral & 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*

We can construct a collection of nonlinear Markov chains 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 distibutions (on $(E_{1:n}, \mathcal{E}_{1:n})$) associated to $\{\pi_n\}$ and $Q_{n,\mu} : \mathcal{P}(E_{1:n-1}) \times E_{n-1} \rightarrow \mathcal{P}(E_{1:n})$. The simplest transition kernel is given by

$$Q_{n,\mu}\left(\mathbf{x}_{1:n-1}, \mathbf{x}'_{1:n}\right) = \Psi_n\left(\mu \times K_n\right)\left(\mathbf{x}'_{1:n}\right) \tag{26}$$

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

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

is a Boltzmann-Gibbs distribution.

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

$$\begin{aligned} Q_{n,\mu}(\mathbf{x}_{1:n-1}, \mathbf{x}'_{1:n}) = {} & \frac{\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}'_n)}{C_n} \mathbb{I}_{\mathbf{x}_{1:n-1}}\left(\mathbf{x}'_{1:n-1}\right) K_n(\mathbf{x}'_{n-1}, \mathbf{x}'_n) + \Bigg(1 - \\ & \int_{E_{1:n}} \frac{\widetilde{w}_n(\mathbf{x}_{n-1}, \mathbf{x}'_n)}{C_n} \delta_{\mathbf{x}_{1:n-1}}\left(d\mathbf{x}'_{1:n-1}\right) K_n(\mathbf{x}'_{n-1}, d\mathbf{x}'_n)\Bigg) \times \\ & \Psi_n(\mu \times K_n)(\mathbf{x}'_{1:n}). \end{aligned} \tag{27}$$

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 $\frac{\widetilde{w}_n(\mathbf{x}_{n-1:n})}{C_n}$ 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 (25) given here by (26) or (27). The SMC algorithm described in Section 2 can be interpreted as a simple Monte Carlo implementation of (26). Whenever $\widetilde{w}_n\left(\mathbf{x}_{n-1}, \mathbf{x}_n\right) \leq C_n$, it is also possible to approximate (27) instead. Under regularity assumptions, it can be shown that this alternative Monte Carlo approximation has a lower asymptotic variance than (26) 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 $X_{1,1}^{(i)} \sim \widetilde{\pi}_1$ and set $\widetilde{\pi}_1^i = \left(1 - \frac{1}{i}\right)\widetilde{\pi}_1^{i-1} + \frac{1}{i}\delta_{X_{1,1}^{(i)}}$.
- **At time** n; $n = 2, ..., P$. Sample $X_{n,1:n}^{(i)} \sim Q_{n,\widetilde{\pi}_{n-1}^i}\left(X_{n-1,1:n-1}^{(i)}, \cdot\right)$ and set $\widetilde{\pi}_n^i = \left(1 - \frac{1}{i}\right)\widetilde{\pi}_n^{i-1} + \frac{1}{i}\delta_{X_{n,1:n}^{(i)}}$.

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 $\left\{X_{n,n-1:n}^{(i)}\right\}$ 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 & Miclo (2004; 2006) and Andrieu et al. (2006) and Brockwell & 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 (5) given by

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

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 (6) as

$$w_n(\mathbf{x}_n) = w_{n-1}(\mathbf{x}_{n-1})\, p(y_n | x_{n-1}) \propto w_{n-1}(\mathbf{x}_{n-1}) \frac{p(\mathbf{x}_{n-1} | y_{1:n})}{p(\mathbf{x}_{n-1} | y_{1:n-1})} \tag{28}$$

If one can sample from $p(x_n | y_n, x_{n-1})$ but cannot compute (28) 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

$$p\left(y_n|\,x_{n-1}\right)=\int\widehat{f}\left(x_n|\,x_{n-1}\right)\widehat{g}\left(y_n|\,x_n\right)dx_n \\ \times\int\frac{f\left(x_n|\,x_{n-1}\right)g\left(y_n|\,x_n\right)}{\widehat{f}\left(x_n|\,x_{n-1}\right)\widehat{g}\left(y_n|\,x_n\right)}\widehat{p}\left(x_n|\,y_n,x_{n-1}\right)dx_n.$$

An alternative consists of moving using $\widehat{p}\left(x_n|\,y_n,x_{n-1}\right)$ -see (9)- and computing the weights using (11)

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

We want to emphasize that such sampling strategies can perform poorly even if one can sample from $p\left(x_n|\,y_n,x_{n-1}\right)$ and compute exactly the associated importance weight. Indeed, in situations where the discrepancy between $p\left(\mathbf{x}_{n-1}|\,y_{1:n-1}\right)$ and $p\left(\mathbf{x}_{n-1}|\,y_{1:n}\right)$ is high, then the weights (28) 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 $\mathbf{x}_{n-1}=\left(\mathbf{u}_{n-1},\mathbf{v}_{n-1}\right)=\left(x_{1:n-R},x_{n-R+1:n-1}\right)$ to $\mathbf{x}_n=\left(\mathbf{u}_{n-1},\mathbf{v}_n\right)=\left(x_{1:n-R},x'_{n-R+1:n}\right)$ and the conditionally optimal distribution is given by

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

Although attractive, this strategy is difficult to apply, as sampling from $p\left(x'_{n-R+1:n}\middle|\,y_{n-R+1:n},x_{n-R}\right)$ becomes more difficult as R increases. Moreover, it requires the ability to compute or obtain unbiased estimates of both $p\left(y_{n-R+1:n}|\,x_{n-R}\right)$ and $1/\eta_{n-1}\left(x_{1:n-R}\right)$ to calculate (6). If we use an approximation $\widehat{\pi}_n\left(\mathbf{v}_n|\,\mathbf{u}_{n-1}\right)$ of $\pi_n\left(\mathbf{v}_n|\,\mathbf{u}_{n-1}\right)$ to move the particles, it remains difficult to compute (10) as we still require an unbiased estimate of $1/\eta_{n-1}$ $\left(x_{1:n-R}\right)$. The discussion of Section 2.3.2 indicates that, alternatively, we can simply weight the particles sampled using $\widehat{\pi}_n\left(\mathbf{v}_n|\,\mathbf{u}_{n-1}\right)$ by (19); this only requires us being able to derive an approximation of $\pi_{n-1}\left(\mathbf{v}_{n-1}|\,\mathbf{u}_{n-1}\right)$.

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 $\widehat{\pi}_n(\mathbf{v}_n|\,\mathbf{u}_{n-1})$ (resp. $\widehat{\pi}_{n-1}(\mathbf{v}_{n-1}|\,\mathbf{u}_{n-1})$) of $\pi_n(\mathbf{v}_n|\,\mathbf{u}_{n-1})$ (resp. $\widehat{\pi}_{n-1}(\mathbf{v}_{n-1}|\,\mathbf{u}_{n-1})$) 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 $p(x_k|\,y_k, x_{k-1}, x_{k+1})$ 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.

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$^{\text{nd}}$ 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 $p(x_1|\,y_{1:100})$ remains very large whereas it is close to 1 for standard methods.

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

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

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 & 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 & Chib 1993). It is well known (e.g. Holmes & 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 (17) can greatly improve the performance of Resample Move (Chopin, 2002; Gilks & Berzuini, 2001) based on (20).

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.:

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

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 & 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 & 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 & Held (2006)), but we only consider the method of Albert & 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.

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:

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

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 (20) and so the incremental weight is 1). For the corresponding backward kernel, L_{n-1}, we consider two options (20) and (17). Since (17) 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, (17) provides substantial improvements over the reversal kernel (20) 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).

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 (17) provided significant improvements over Resample Move. This is of interest when the full conditionals are not available, but good approxmations 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 (20). We note that this is not meaningless

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

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.

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 (a likely) lower CPU time.

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 & Shepard 2001) found in Roberts et al. (2004). That is, the price 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:

$$d\sigma_t = -\mu\sigma_t dt + dZ_t \tag{29}$$

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 & Shephard 2001; Applebaum 2004) that for any self-decomposable random variable, there exists a unique Lévy process that satisfies (29); 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

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

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):

$$\sigma_0|\theta, \nu \sim \mathcal{G}a(\nu, \theta),\ \nu \sim \mathcal{G}a(\alpha_\nu, \beta_\nu),$$
$$\mu \sim \mathcal{G}a(\alpha_\mu, \beta_\mu),\ \ \theta \sim \mathcal{G}a(\alpha_\theta, \beta_\theta)$$

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

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

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:

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

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:

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

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.

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.

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 & Berzuini (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.

It should be noted 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 & 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. & 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. Technical report, Department of Mathematics, University of Bristol.

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

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

Brockwell, A.E. and Doucet, A. (2006) Sequentially interacting Markov chain Monte Carlo for Bayesian computation, Technical Report, Department of Statistics, Carnegie Mellon University.

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. Technical report Cambridge University, CUED/F-INFENG/TR381.

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

Del Moral, P. and Doucet, A. (2003) On a class of genealogical and interacting Metropolis models. In *Séminaire de Probabilités XXXVII*, Ed. Azéma, J., Emery, M., Ledoux, M. and Yor, M., *Lecture Notes in Mathematics*, Berlin: Springer, **1832,** 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 Appli.*, vol. 3, 615-660.

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

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. & 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.*, Vol.16 No.2, pp.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. Technical report, Department of Mathematics, Imperial College London.

Jasra, A., Stephens, D. A. and Holmes, C.C. (2005b) On population-based simulation for static inference. Technical report, Department of Mathematics, Imperial College London.

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

Neal, R. (1996) Sampling from multimodal distributions via tempered transitions. 6, 353-366.

Neal, R. (2001) Annealed importance sampling. 11, 125-139.

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

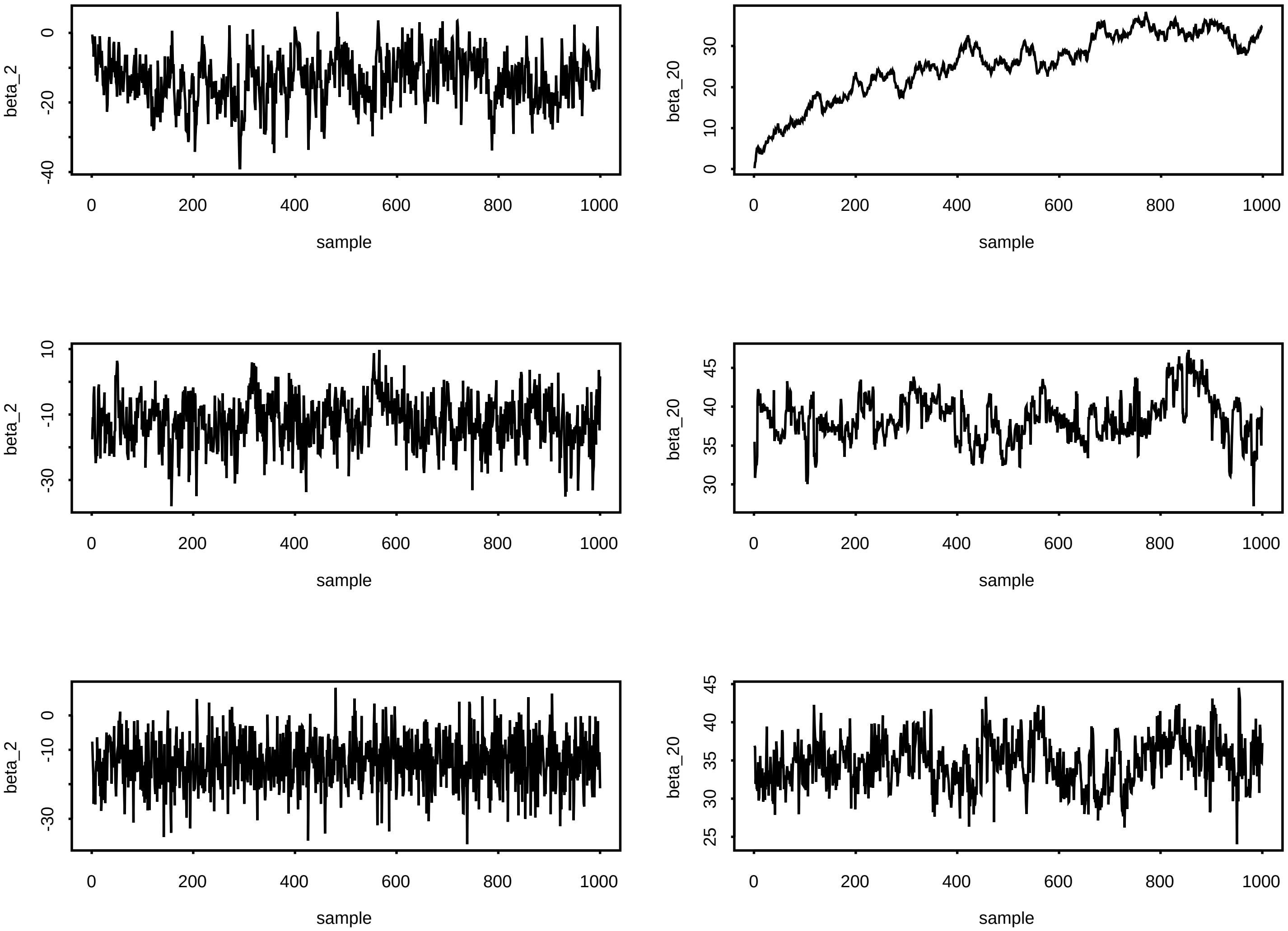

Figure 1: *Sampled coefficients from the binary regression example. For the MCMC (top row), we ran the Gibbs sampler of Albert & Chib (1993) for 100000 iterations and stored every 100th (CPU time 421 sec). For the reversal SMC (middle row) we ran 1000 particles for 200 time steps (CPU 681 sec), the final ESS was 790. For the Gibbs SMC (bottom row) we did the same except the CPU was 677 and the ESS was 557.*

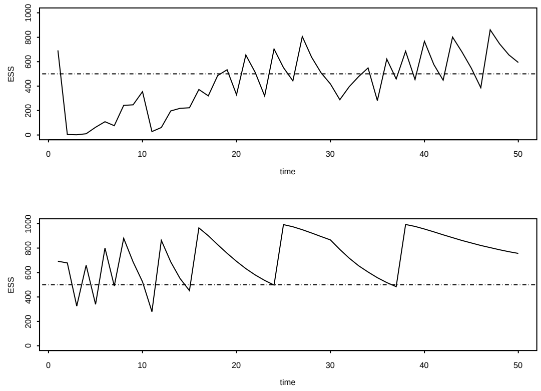

Figure 2: *ESS plots from the binary regression example; 50 time points. The top graph is for reversal kernel (17). We sampled 1000 particles and resampled when the ESS dropped below 500 particles,*

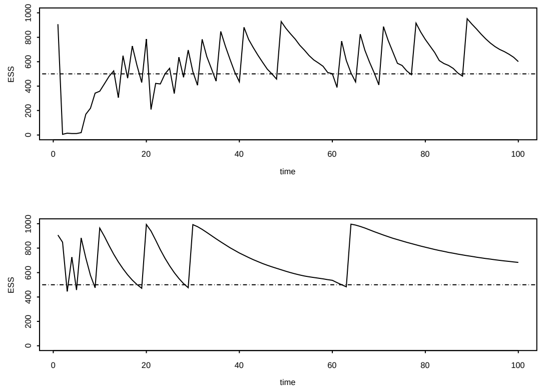

Figure 3: *ESS plots from the binary regression example; 100 time points. The top graph is for reversal kernel (17). We sampled 1000 particles and resampled when the ESS dropped below 500 particles,*

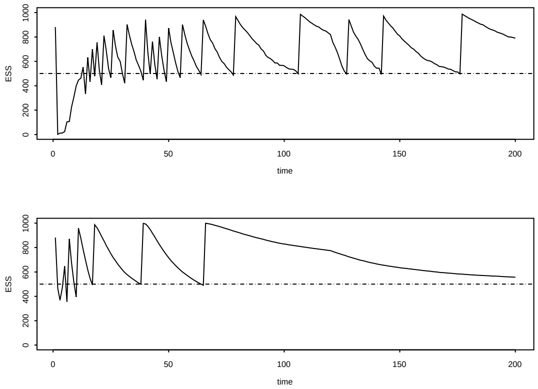

Figure 4: *ESS plots from the binary regression example; 200 time points. The top graph is for reversal kernel (17). We sampled 1000 particles and resampled when the ESS dropped below 500 particles,*

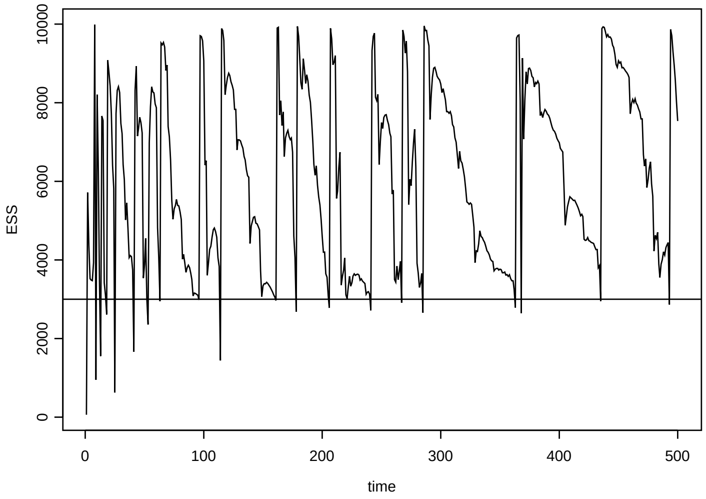

Figure 5: *ESS plot for simulated data from the stochastic volatility example. We ran 10000 particles with resampling threshold (−−) 3000 particles.*

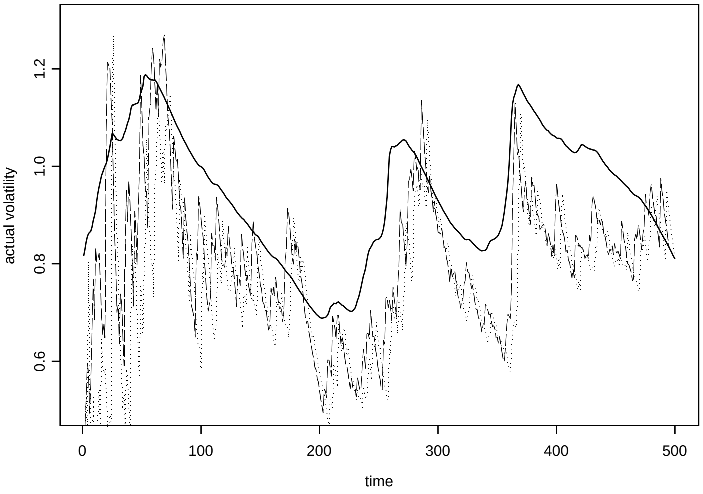

Figure 6: *Actual volatility for simulated data from the stochastic volatility example. We plotted the actual volatility for the final density (full line) filtered (esimated at each timepoint, dot) and smoothed (estimated at each timepoint, lag 5, dash)*