A Tutorial on Particle Filtering and Smoothing: Fifteen years later

Arnaud Doucet
The Institute of Statistical Mathematics,
4-6-7 Minami-Azabu, Minato-ku,
Tokyo 106-8569, Japan.
Email: Arnaud@ism.ac.jp

Adam M. Johansen
Department of Statistics,
University of Warwick,
Coventry, CV4 7AL, UK
Email: A.M.Johansen@warwick.ac.uk

First Version 1.0 – April 2008
This Version 1.1 – December 2008

Abstract

Optimal estimation problems for non-linear non-Gaussian state-space models do not typically admit analytic solutions. Since their introduction in 1993, particle filtering methods have become a very popular class of algorithms to solve these estimation problems numerically in an online manner, i.e. recursively as observations become available, and are now routinely used in fields as diverse as computer vision, econometrics, robotics and navigation. The objective of this tutorial is to provide a complete, up-to-date survey of this field as of 2008. Basic and advanced particle methods for filtering as well as smoothing are presented.

Keywords: Central Limit Theorem, Filtering, Hidden Markov Models, Markov chain Monte Carlo, Particle methods, Resampling, Sequential Monte Carlo, Smoothing, State-Space models.

1 Introduction

The general state space hidden Markov models, which are summarised in section 2.1, provide an extremely flexible framework for modelling time series. The great descriptive power of these models comes at the expense of intractability: it is impossible to obtain analytic solutions to the inference problems of interest with the exception of a small number of particularly simple cases. The "particle" methods described by this tutorial are a broad and popular class of Monte Carlo algorithms which have been developed over the past fifteen years to provide approximate solutions to these intractable inference problems.

1.1 Preliminary remarks

Since their introduction in 1993 [22], particle filters have become a very popular class of numerical methods for the solution of optimal estimation problems in non-linear non-Gaussian scenarios. In comparison with standard approximation methods, such as the popular Extended Kalman Filter, the principal advantage of particle methods is that they do not rely on any local linearisation technique or any crude functional approximation. The price that must be paid for this flexibility is computational: these methods are computationally expensive. However, thanks to the availability of ever-increasing computational power, these methods are already used in real-time applications appearing in fields as diverse as chemical engineering, computer vision, financial econometrics, target tracking and robotics. Moreover, even in scenarios in which there are no real-time constraints, these methods can be a powerful alternative to Markov chain Monte Carlo (MCMC) algorithms — alternatively, they can be used to design very efficient MCMC schemes.

As a result of the popularity of particle methods, a few tutorials have already been published on the subject [3, 8, 18, 29]. The most popular, [3], dates back to 2002 and, like the edited volume [16] from 2001, it is now somewhat outdated. This tutorial differs from previously published tutorials in two ways. First, the obvious: it is, as of December 2008, the most recent tutorial on the subject and so it has been possible to include some very recent material on advanced particle methods for filtering and smoothing. Second, more importantly, this tutorial was not intended to resemble a cookbook. To this end, all of the algorithms are presented within a simple, unified framework. In particular, we show that essentially all basic and advanced methods for particle filtering can be reinterpreted as some special instances of a single generic Sequential Monte Carlo (SMC) algorithm. In our opinion, this framework is not only elegant but allows the development of a better intuitive and theoretical understanding of particle methods. It also shows that essentially any particle filter can be implemented using a simple computational framework such as that provided by [24]. Absolute beginners might benefit from reading [17], which provides an elementary introduction to the field, before the present tutorial.

1.2 Organisation of the tutorial

The rest of this paper is organised as follows. In Section 2, we present hidden Markov models and the associated Bayesian recursions for the filtering and smoothing distributions. In Section 3, we introduce a generic SMC algorithm which provides weighted samples from any sequence of probability distributions. In Section 4, we show how all the (basic and advanced) particle filtering methods developed in the literature can be interpreted as special instances of the generic SMC algorithm presented in Section 3. Section 5 is devoted to particle smoothing and we mention some open problems in Section 6.

2 Bayesian Inference in Hidden Markov Models

2.1 Hidden Markov Models and Inference Aims

Consider an $\mathcal{X}-$valued discrete-time Markov process $\{X_n\}_{n\geq 1}$ such that

$$X_1 \sim \mu(x_1) \text{ and } X_n|(X_{n-1}=x_{n-1}) \sim f(x_n|x_{n-1}) \tag{1}$$

where "$\sim$" means distributed according to, $\mu(x)$ is a probability density function and $f(x|x')$ denotes the probability density associated with moving from x' to x. All the densities are with respect to a dominating measure that we will denote, with abuse of notation, dx. We are interested in estimating $\{X_n\}_{n\geq 1}$ but only have access to the $\mathcal{Y}-$valued process $\{Y_n\}_{n\geq 1}$. We assume that, given $\{X_n\}_{n\geq 1}$, the observations $\{Y_n\}_{n\geq 1}$ are statistically independent and their marginal densities (with respect to a dominating measure dy_n) are given by

$$Y_n|(X_n=x_n) \sim g(y_n|x_n). \tag{2}$$

For the sake of simplicity, we have considered only the homogeneous case here; that is, the transition and observation densities are independent of the time index n. The extension to the inhomogeneous case is straightforward. It is assumed throughout that any model parameters are known.

Models compatible with (1)-(2) are known as hidden Markov models (HMM) or general state-space models (SSM). This class includes many models of interest. The following examples provide an illustration of several simple problems which can be dealt with within this framework. More complicated scenarios can also be considered.

Example 1 - *Finite State-Space HMM.* In this case, we have $\mathcal{X}=\{1,...,K\}$ so

$$\Pr(X_1=k)=\mu(k), \quad \Pr(X_n=k|X_{n-1}=l)=f(k|l).$$

The observations follow an arbitrary model of the form (2). This type of model is extremely general and examples can be found in areas such as genetics in which they can describe imperfectly observed genetic sequences, signal processing, and computer science in which they can describe, amongst many other things, arbitrary finite-state machines.

Example 2 - *Linear Gaussian model.* Here, $\mathcal{X}=\mathbb{R}^{n_x}$, $\mathcal{Y}=\mathbb{R}^{n_y}$, $X_1 \sim \mathcal{N}(0,\Sigma)$ and

$$\begin{aligned} X_n &= AX_{n-1}+BV_n, \\ Y_n &= CX_n+DW_n \end{aligned}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,I_{n_v})$, $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,I_{n_w})$ and A,B,C,D are matrices of appropriate dimensions. Note that $\mathcal{N}(m,\Sigma)$ denotes a Gaussian distribution of mean m and variance-covariance matrix Σ, whereas $\mathcal{N}(x;m,\Sigma)$ denotes the Gaussian density of argument x and similar statistics. In this case $\mu(x)=\mathcal{N}(x;0,\Sigma)$, $f(x'|x)=\mathcal{N}\left(x';Ax,BB^{\mathrm{T}}\right)$ and $g(y|x)=\mathcal{N}\left(y;Cx,DD^{\mathrm{T}}\right)$. As inference is analytically tractable for this model, it has been extremely widely used for problems such as target tracking and signal processing.

Example 3 - *Switching Linear Gaussian model.* We have $\mathcal{X}=\mathcal{U}\times\mathcal{Z}$ with $\mathcal{U}=\{1,...,K\}$ and $\mathcal{Z}=\mathbb{R}^{n_z}$. Here $X_n=(U_n,Z_n)$ where $\{U_n\}$ is a finite state-space Markov chain such that $\Pr(U_1=k)=\mu_U(k)$, $\Pr(U_n=k|U_{n-1}=l)=f_U(k|l)$ and conditional upon $\{U_n\}$ we have a linear Gaussian model with $Z_1|U_1 \sim \mathcal{N}(0,\Sigma_{U_1})$ and

$$\begin{aligned} Z_n &= A_{U_n}Z_{n-1}+B_{U_n}V_n, \\ Y_n &= C_{U_n}Z_n+D_{U_n}W_n \end{aligned}$$

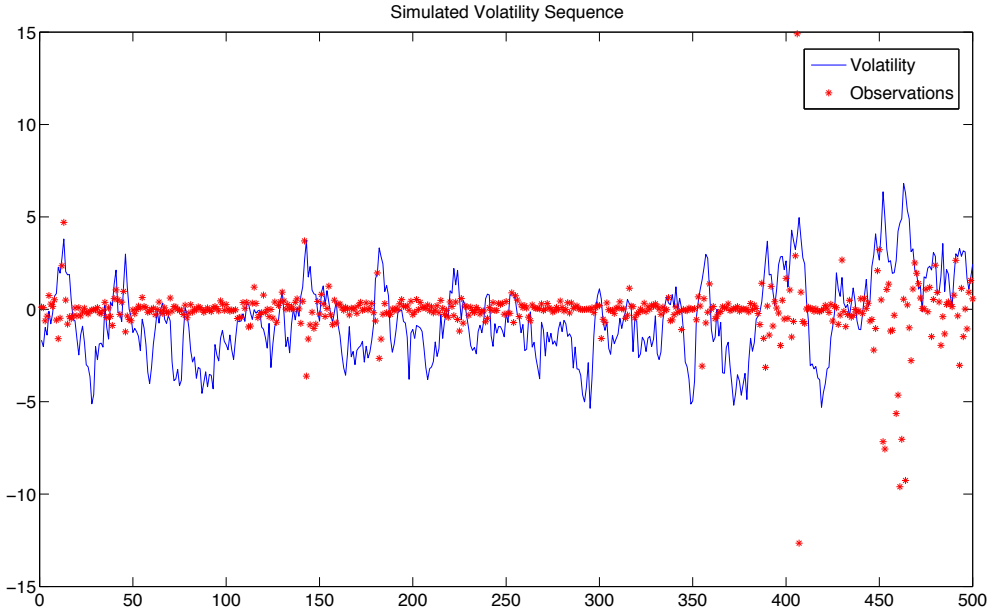

Figure 1: A simulation of the stochastic volatility model described in example 4.

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{n_v})$, $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{n_w})$ and $\{A_k, B_k, C_k, D_k; k = 1, ..., K\}$ are matrices of appropriate dimensions. In this case we have $\mu(x) = \mu(u, z) = \mu_U(u)\mathcal{N}(z; 0, \Sigma_u)$, $f(x'|x) = f((u', z')|(u, z)) = f_U(u'|u)\mathcal{N}\left(z'; A_{u'}z, B_{u'}B_{u'}^{\mathrm{T}}\right)$ and $g(y|x) = g(y|(u, z)) = \mathcal{N}\left(y; C_u z, D_u D_u^{\mathrm{T}}\right)$. This type of model provides a generalisation of that described in example 2 with only a slight increase in complexity.

Example 4 - *Stochastic Volatility model.* We have $\mathcal{X} = \mathcal{Y} = \mathbb{R}$, $X_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\alpha^2}\right)$ and

$$\begin{aligned} X_n &= \alpha X_{n-1} + \sigma V_n, \\ Y_n &= \beta \exp(X_n/2) W_n \end{aligned}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. In this case we have $\mu(x) = \mathcal{N}\left(x; 0, \frac{\sigma^2}{1-\alpha^2}\right)$, $f(x'|x) = \mathcal{N}\left(x'; \alpha x, \sigma^2\right)$ and $g(y|x) = \mathcal{N}\left(y; 0, \beta^2 \exp(x)\right)$. Note that this choice of initial distribution ensures that the marginal distribution of X_n is also $\mu(x)$ for all n. This type of model, and its generalisations, have been very widely used in various areas of economics and mathematical finance: inferring and predicting underlying volatility from observed price or rate data is an important problem. Figure 1 shows a short section of data simulated from such a model with parameter values $\alpha = 0.91, \sigma = 1.0$ and $\beta = 0.5$ which will be used below to illustrate the behaviour of some simple algorithms.

Equations (1)-(2) define a Bayesian model in which (1) defines the prior distribution of the process of interest $\{X_n\}_{n\geq 1}$ and (2) defines the likelihood function; that is:

$$p(x_{1:n}) = \mu(x_1) \prod_{k=2}^{n} f(x_k|x_{k-1}) \tag{3}$$

and

$$p(y_{1:n}|x_{1:n}) = \prod_{k=1}^{n} g(y_k|x_k), \tag{4}$$

where, for any sequence $\{z_n\}_{n\geq 1}$, and any $i \leq j$, $z_{i:j} := (z_i, z_{i+1}, ..., z_j)$.

In such a Bayesian context, inference about $X_{1:n}$ given a realisation of the observations $Y_{1:n} = y_{1:n}$ relies upon the posterior distribution

$$p(x_{1:n}|y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})}, \tag{5}$$

where

$$p(x_{1:n}, y_{1:n}) = p(x_{1:n})\, p(y_{1:n}|x_{1:n}), \tag{6}$$

$$\text{and } p(y_{1:n}) = \int p(x_{1:n}, y_{1:n})\, dx_{1:n}. \tag{7}$$

For the finite state-space HMM model discussed in Example 1, the integrals correspond to finite sums and all these (discrete) probability distributions can be computed exactly. For the linear Gaussian model discussed in Example 2, it is easy to check that $p(x_{1:n}|y_{1:n})$ is a Gaussian distribution whose mean and covariance can be computed using Kalman techniques; see [1], for example. However, for most non-linear non-Gaussian models, it is not possible to compute these distributions in closed-form and we need to employ numerical methods. Particle methods are a set of flexible and powerful simulation-based methods which provide samples approximately distributed according to posterior distributions of the form $p(x_{1:n}|y_{1:n})$ and facilitate the approximate calculation of $p(y_{1:n})$. Such methods are a subset of the class of methods known as Sequential Monte Carlo (SMC) methods.

In this tutorial, we will review various particle methods to address the following problems:

• *Filtering and Marginal likelihood computation*: Assume that we are interested in the sequential approximation of the distributions $\{p(x_{1:n}|y_{1:n})\}_{n\geq 1}$ and marginal likelihoods $\{p(y_{1:n})\}_{n\geq 1}$. That is, we wish to approximate $p(x_1|y_1)$ and $p(y_1)$ at the first time instance, $p(x_{1:2}|y_{1:2})$ and $p(y_{1:2})$ at the second time instance and so on. We will refer to this problem as the optimal filtering problem. This is slightly at variance with the usage in much of the literature which reserves the term for the estimation of the marginal distributions $\{p(x_n|y_{1:n})\}_{n\geq 1}$ rather than the joint distributions $\{p(x_{1:n}|y_{1:n})\}_{n\geq 1}$.

We will describe basic and advanced particle filtering methods to address this problem including auxiliary particle filtering, particle filtering with MCMC moves, block sampling strategies and Rao-Blackwellised particle filters.

• *Smoothing*: Consider attempting to sample from a joint distribution $p(x_{1:T}|y_{1:T})$ and approximating the associated marginals $\{p(x_n|y_{1:T})\}$ where $n = 1, ..., T$. Particle filtering techniques can be used to solve this problem but perform poorly when T is large for reasons detailed in this tutorial. We will describe several particle smoothing methods to address this problem. Essentially, these methods rely on the particle implementation of the forward filtering-backward smoothing formula or of a generalised version of the two-filter smoothing formula.

2.2 Filtering and Marginal Likelihood

The first area of interest, and that to which the vast majority of the literature on particle methods has been dedicated from the outset, is the problem of filtering: characterising the distribution of the state of the hidden Markov model at the present time, given the information provided by all of the observations received up to the present time. This can be thought of as a "tracking" problem: keeping track of the current "location" of the system given noisy observations — and, indeed, this is an extremely popular area of application for these methods. The term is sometimes also used to refer to the practice of estimating the full trajectory of the state sequence up to the present time given the observations received up to this time.

We recall that, following (1)-(2), the posterior distribution $p(x_{1:n}|y_{1:n})$ is defined by (5) — the prior is defined in (3) and the likelihood in (4). The unnormalised posterior distribution $p(x_{1:n}, y_{1:n})$ given in (5) satisfies

$$p(x_{1:n}, y_{1:n}) = p(x_{1:n-1}, y_{1:n-1})\, f(x_n|x_{n-1})\, g(y_n|x_n). \tag{8}$$

Consequently, the posterior $p(x_{1:n}|y_{1:n})$ satisfies the following recursion

$$p(x_{1:n}|y_{1:n}) = p(x_{1:n-1}|y_{1:n-1}) \frac{f(x_n|x_{n-1})\, g(y_n|x_n)}{p(y_n|y_{1:n-1})}, \tag{9}$$

where

$$p(y_n|y_{1:n-1}) = \int p(x_{n-1}|y_{1:n-1})\, f(x_n|x_{n-1})\, g(y_n|x_n)\, dx_{n-1:n} \tag{10}$$

In the literature, the recursion satisfied by the marginal distribution $p(x_n|y_{1:n})$ is often presented. It is straightforward to check (by integrating out $x_{1:n-1}$ in (9)) that we have

$$p(x_n|y_{1:n}) = \frac{g(y_n|x_n)\, p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})}, \tag{11}$$

where

$$p(x_n|y_{1:n-1}) = \int f(x_n|x_{n-1})\, p(x_{n-1}|y_{1:n-1})\, dx_{n-1}. \tag{12}$$

Equation (12) is known as the prediction step and (11) is known as the updating step. However, most particle filtering methods rely on a numerical approximation of recursion (9) and not of (11)-(12).

If we can compute $\{p(x_{1:n}|y_{1:n})\}$ and thus $\{p(x_n|y_{1:n})\}$ sequentially, then the quantity $p(y_{1:n})$, which is known as the marginal likelihood, can also clearly be evaluated recursively using

$$p(y_{1:n}) = p(y_1) \prod_{k=2}^{n} p(y_k|y_{1:k-1}) \tag{13}$$

where $p(y_k|y_{1:k-1})$ is of the form (10).

2.3 Smoothing

One problem, which is closely related to filtering, but computationally more challenging for reasons which will become apparent later, is known as smoothing. Whereas filtering corresponds to estimating the distribution of the current state of an HMM based upon the observations received up until the current time, smoothing corresponds to estimating the distribution of the state at a particular time given all of the observations up to some *later* time. The trajectory estimates obtained by such methods, as a result of the additional information available, tend to be smoother than those obtained by filtering. It is intuitive that if estimates of the state at time n are not required instantly, then better estimation performance is likely to be obtained by taking advantage of a few later observations. Designing efficient sequential algorithms for the solution of this problem is not quite as straightforward as it might seem, but a number of effective strategies have been developed and are described below.

More formally: assume that we have access to the data $y_{1:T}$, and wish to compute the marginal distributions $\{p(x_n|y_{1:T})\}$ where $n = 1, ..., T$ or to sample from $p(x_{1:T}|y_{1:T})$. In principle, the marginals $\{p(x_n|y_{1:T})\}$ could be obtained directly by considering the joint distribution $p(x_{1:T}|y_{1:T})$ and integrating out the variables $(x_{1:n-1}, x_{n+1:T})$. Extending this reasoning in the context of particle methods, one can simply use the identity $p(x_n|y_{1:T}) = \int p(x_{1:T}|y_{1:T}) dx_{1:n-1} dx_{n+1:T}$ and take the same approach which is used in particle filtering: use Monte Carlo algorithms to obtain an approximate characterisation of the joint distribution and then use the associated marginal distribution to approximate the distributions of interest. Unfortunately, as is detailed below, when $n \ll T$ this strategy is doomed to failure: the marginal distribution $p(x_n|y_{1:n})$ occupies a privileged role within the particle filter framework as it is, in some sense, better characterised than any of the other marginal distributions.

For this reason, it is necessary to develop more sophisticated strategies in order to obtain good smoothing algorithms. There has been much progress in this direction over the past decade. Below, we present two alternative recursions that will prove useful when numerical approximations are required. The key to the success of these recursions is that they rely upon only the marginal filtering distributions $\{p(x_n|y_{1:n})\}$.

2.3.1 Forward-Backward Recursions

The following decomposition of the joint distribution $p(x_{1:T}|y_{1:T})$

$$\begin{aligned} p(x_{1:T}|y_{1:T}) &= p(x_T|y_{1:T}) \prod_{n=1}^{T-1} p(x_n|x_{n+1}, y_{1:T}) \\ &= p(x_T|y_{1:T}) \prod_{n=1}^{T-1} p(x_n|x_{n+1}, y_{1:n}), \end{aligned} \tag{14}$$

shows that, conditional on $y_{1:T}$, $\{X_n\}$ is an inhomogeneous Markov process.

Eq. (14) suggests the following algorithm to sample from $p(x_{1:T}|y_{1:T})$. First compute and store the marginal distributions $\{p(x_n|y_{1:n})\}$ for $n = 1, ..., T$. Then sample $X_T \sim p(x_T|y_{1:T})$ and for $n = T-1, T-2, ..., 1$, sample $X_n \sim p(x_n|X_{n+1}, y_{1:n})$ where

$$p(x_n|x_{n+1}, y_{1:n}) = \frac{f(x_{n+1}|x_n)\, p(x_n|y_{1:n})}{p(x_{n+1}|y_{1:n})}$$

It also follows, by integrating out $(x_{1:n-1}, x_{n+1:T})$ in Eq. (14), that

$$p(x_n|y_{1:T}) = p(x_n|y_{1:n}) \int \frac{f(x_{n+1}|x_n)}{p(x_{n+1}|y_{1:n})} p(x_{n+1}|y_{1:T})\, dx_{n+1}. \tag{15}$$

So to compute $\{p(x_n|y_{1:T})\}$, we simply modify the backward pass and, instead of sampling from $p(x_n|x_{n+1}, y_{1:n})$, we compute $p(x_n|y_{1:T})$ using (15).

2.3.2 Generalised Two-Filter Formula

The two-filter formula is a well-established alternative to the forward-filtering backward-smoothing technique to compute the marginal distributions $\{p(x_n|y_{1:T})\}$ [4]. It relies on the following identity

$$p(x_n|y_{1:T}) = \frac{p(x_n|y_{1:n-1})\, p(y_{n:T}|x_n)}{p(y_{n:T}|y_{1:n-1})},$$

where the so-called backward information filter is initialised at time $n = T$ by $p(y_T|x_T) = g(y_T|x_T)$ and satisfies

$$\begin{aligned} p(y_{n:T}|x_n) &= \int \prod_{k=n+1}^{T} f(x_k|x_{k-1}) \prod_{k=n}^{T} g(y_k|x_k)\, dx_{n+1:T} \\ &= g(y_n|x_n) \int f(x_{n+1}|x_n)\, p(y_{n+1:T}|x_{n+1})\, dx_{n+1}. \end{aligned} \tag{16}$$

The backward information filter is not a probability density in argument x_n and it is even possible that $\int p(y_{n:T}|x_n)\, dx_n = \infty$. Although this is not an issue when $p(y_{n:T}|x_n)$ can be computed exactly, it does preclude the direct use of SMC methods to estimate this integral. To address this problem, a generalised version of the two-filter formula was proposed in [5]. It relies on the introduction of a set of artificial probability distributions $\{\widetilde{p}_n(x_n)\}$ and the joint distributions

$$\widetilde{p}_n(x_{n:T}|y_{n:T}) \propto \widetilde{p}_n(x_n) \prod_{k=n+1}^{T} f(x_k|x_{k-1}) \prod_{k=n}^{T} g(y_k|x_k), \tag{17}$$

which are constructed such that their marginal distributions, $\widetilde{p}_n(x_n|y_{n:T}) \propto \widetilde{p}_n(x_n)\, p(y_{n:T}|x_n)$, are simply "integrable versions" of the backward information filter. It is easy to establish the generalised two-filter formula

$$p(x_1|y_{1:T}) \propto \frac{\mu(x_1)\, \widetilde{p}(x_1|y_{1:T})}{\widetilde{p}_1(x_1)},\ p(x_n|y_{1:T}) \propto \frac{p(x_n|y_{1:n-1})\widetilde{p}(x_n|y_{n:T})}{\widetilde{p}_n(x_n)} \tag{18}$$

which is valid whenever the support of $\widetilde{p}_n(x_n)$ includes the support of the prior $p_n(x_n)$; that is

$$p_n(x_n) = \int \mu(x_1) \prod_{k=2}^{n} f(x_k|x_{k-1})\, dx_{1:n-1} > 0 \Rightarrow \widetilde{p}_n(x_n) > 0.$$

The generalised two-filter smoother for $\{p(x_n|y_{n:T})\}$ proceeds as follows. Using the standard forward recursion, we can compute and store the marginal distributions $\{p(x_n|y_{1:n-1})\}$. Using a backward recursion, we compute and store $\{\widetilde{p}(x_n|y_{n:T})\}$. Then for any $n = 1, ..., T$ we can combine $p(x_n|y_{1:n-1})$ and $\widetilde{p}(x_n|y_{n:T})$ to obtain $p(x_n|y_{1:T})$.

In [4], this identity is discussed in the particular case where $\widetilde{p}_n(x_n) = p_n(x_n)$. However, when computing $\{\widetilde{p}(x_n|y_{n:T})\}$ using SMC, it is necessary to be able to compute $\widetilde{p}_n(x_n)$ exactly hence this rules out the choice $\widetilde{p}_n(x_n) = p_n(x_n)$ for most non-linear non-Gaussian models. In practice, we should select a heavy-tailed approximation of $p_n(x_n)$ for $\widetilde{p}_n(x_n)$ in such settings. It is also possible to use the generalised-two filter formula to sample from $p(x_{1:T}|y_{1:T})$; see [5] for details.

2.4 Summary

Bayesian inference in non-linear non-Gaussian dynamic models relies on the sequence of posterior distributions $\{p(x_{1:n}|y_{1:n})\}$ and its marginals. Except in simple problems such as Examples 1 and 2, it is not possible to compute these distributions in closed-form. In some scenarios, it might be possible to obtain reasonable performance by employing functional approximations of these distributions. Here, we will discuss only Monte Carlo approximations of these distributions; that is numerical schemes in which the distributions of interest are approximated by a large collection of N random samples termed particles. The main advantage of such methods is that under weak assumptions they provide asymptotically (i.e. as $N \to \infty$) consistent estimates of the target distributions of interest. It is also noteworthy that these techniques can be applied to problems of moderately-high dimension in which traditional numerical integration might be expected to perform poorly.

3 Sequential Monte Carlo Methods

Over the past fifteen years, particle methods for filtering and smoothing have been the most common examples of SMC algorithms. Indeed, it has become traditional to present particle filtering and SMC as being the same thing in much of the literature. Here, we wish to emphasise that SMC actually encompasses a broader range of algorithms — and by doing so we are able to show that many more advanced techniques for approximate filtering and smoothing can be described using precisely the same framework and terminology as the basic algorithm.

SMC methods are a general class of Monte Carlo methods that sample sequentially from a sequence of target probability densities $\{\pi_n(x_{1:n})\}$ of increasing dimension where each distribution $\pi_n(x_{1:n})$ is defined on the product space $\mathcal{X}^n$. Writing

$$\pi_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{Z_n} \tag{19}$$

we require only that $\gamma_n : \mathcal{X}^n \to \mathbb{R}^+$ is known pointwise; the normalising constant

$$Z_n = \int \gamma_n(x_{1:n})\, dx_{1:n} \tag{20}$$

might be unknown. SMC provide an approximation of $\pi_1(x_1)$ and an estimate of Z_1 at time 1 then an approximation of $\pi_2(x_{1:2})$ and an estimate of Z_2 at time 2 and so on.

For example, in the context of filtering, we could have $\gamma_n(x_{1:n}) = p(x_{1:n}, y_{1:n})$, $Z_n = p(y_{1:n})$ so $\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n})$. However, we emphasise that this is just one particular choice of target distributions. Not only can SMC methods be used outside the filtering context but, more importantly for this tutorial, some advanced particle filtering and smoothing methods discussed below do not rely on this sequence of target distributions. Consequently, we believe that understanding the main principles behind generic SMC methods is essential to the development of a proper understanding of particle filtering and smoothing methods.

We start this section with a very basic review of Monte Carlo methods and Importance Sampling (IS). We then present the Sequential Importance Sampling (SIS) method, point out the limitations of this method and show how resampling techniques can be used to partially mitigate them. Having introduced the basic particle filter as an SMC method, we show how various advanced techniques which have been developed over the past fifteen years can themselves be interpreted within the same formalism as SMC algorithms associated with sequences of distributions which may not coincide with the filtering distributions. These alternative sequences of target distributions are either constructed such that they admit the distributions $\{p(x_{1:n}|y_{1:n})\}$ as marginal distributions, or an importance sampling correction is necessary to ensure the consistency of estimates.

3.1 Basics of Monte Carlo Methods

Initially, consider approximating a generic probability density $\pi_n(x_{1:n})$ for some fixed n. If we sample N independent random variables, $X_{1:n}^i \sim \pi_n(x_{1:n})$ for $i = 1, ..., N$, then the Monte Carlo method approximates $\pi_n(x_{1:n})$ by the empirical measure[1]

$$\widehat{\pi}_n(x_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:n}^i}(x_{1:n}),$$

where $\delta_{x_0}(x)$ denotes the Dirac delta mass located at x_0. Based on this approximation, it is possible to approximate any marginal, say $\pi_n(x_k)$, easily using

$$\widehat{\pi}_n(x_k) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_k^i}(x_k),$$

and the expectation of any test function $\varphi_n : \mathcal{X}^n \to \mathbb{R}$ given by

$$I_n(\varphi_n) := \int \varphi_n(x_{1:n})\, \pi_n(x_{1:n})\, dx_{1:n},$$

is estimated by

$$I_n^{\mathrm{MC}}(\varphi_n) := \int \varphi_n(x_{1:n})\, \widehat{\pi}_n(x_{1:n})\, dx_{1:n} = \frac{1}{N}\sum_{i=1}^{N} \varphi_n\left(X_{1:n}^i\right).$$

It is easy to check that this estimate is unbiased and that its variance is given by

$$\mathbb{V}\left[I_n^{\mathrm{MC}}(\varphi_n)\right] = \frac{1}{N}\left(\int \varphi_n^2(x_{1:n})\, \pi_n(x_{1:n})\, dx_{1:n} - I_n^2(\varphi_n)\right).$$

The main advantage of Monte Carlo methods over standard approximation techniques is that the variance of the approximation error decreases at a rate of $\mathcal{O}(1/N)$ regardless of the dimension of the space $\mathcal{X}^n$. However, there are at least two main problems with this basic Monte Carlo approach:

• *Problem 1*: If $\pi_n(x_{1:n})$ is a complex high-dimensional probability distribution, then we cannot sample from it.

[1] We persist with the abusive use of density notation in the interests of simplicity and accessibility; the alternations required to obtain a rigorous formulation are obvious.

• *Problem 2*: Even if we knew how to sample exactly from $\pi_n(x_{1:n})$, the computational complexity of such a sampling scheme is typically at least linear in the number of variables n. So an algorithm sampling exactly from $\pi_n(x_{1:n})$, sequentially for each value of n, would have a computational complexity increasing at least linearly with n.

3.2 Importance Sampling

We are going to address *Problem 1* using the IS method. This is a fundamental Monte Carlo method and the basis of all the algorithms developed later on. IS relies on the introduction of an *importance density*[2] $q_n(x_{1:n})$ such that

$$\pi_n(x_{1:n}) > 0 \Rightarrow q_n(x_{1:n}) > 0.$$

In this case, we have from (19)-(20) the following IS identities

$$\pi_n(x_{1:n}) = \frac{w_n(x_{1:n})\, q_n(x_{1:n})}{Z_n}, \tag{21}$$

$$Z_n = \int w_n(x_{1:n})\, q_n(x_{1:n})\, dx_{1:n} \tag{22}$$

where $w_n(x_{1:n})$ is the *unnormalised weight* function

$$w_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{q_n(x_{1:n})}.$$

In particular, we can select an importance density $q_n(x_{1:n})$ from which it is easy to draw samples; e.g. a multivariate Gaussian. Assume we draw N independent samples $X_{1:n}^i \sim q_n(x_{1:n})$ then by inserting the Monte Carlo approximation of $q_n(x_{1:n})$ — that is the empirical measure of the samples $X_{1:n}^i$ — into (21)–(22) we obtain

$$\widehat{\pi}_n(x_{1:n}) = \sum_{i=1}^{N} W_n^i \delta_{X_{1:n}^i}(x_{1:n}), \tag{23}$$

$$\widehat{Z}_n = \frac{1}{N}\sum_{i=1}^{N} w_n\left(X_{1:n}^i\right) \tag{24}$$

where

$$W_n^i = \frac{w_n\left(X_{1:n}^i\right)}{\sum_{j=1}^{N} w_n\left(X_{1:n}^j\right)}. \tag{25}$$

Compared to standard Monte Carlo, IS provides an (unbiased) estimate of the normalising constant with relative variance

$$\frac{\mathbb{V}_{\mathrm{IS}}\left[\widehat{Z}_n\right]}{Z_n^2} = \frac{1}{N}\left(\int \frac{\pi_n^2(x_{1:n})}{q_n(x_{1:n})} dx_{1:n} - 1\right). \tag{26}$$

If we are interested in computing $I_n(\varphi_n)$, we can also use the estimate

$$I_n^{\mathrm{IS}}(\varphi_n) = \int \varphi_n(x_{1:n})\, \widehat{\pi}_n(x_{1:n})\, dx_{1:n} = \sum_{i=1}^{N} W_n^i \varphi_n\left(X_{1:n}^i\right).$$

Unlike $I_n^{\mathrm{MC}}(\varphi_n)$, this estimate is biased for finite N. However, it is consistent and it is easy to check that its asymptotic bias is given by

$$\lim_{N\to\infty} N\left(I_n^{\mathrm{IS}}(\varphi_n) - I_n(\varphi_n)\right) = -\int \frac{\pi_n^2(x_{1:n})}{q_n(x_{1:n})}\left(\varphi_n(x_{1:n}) - I_n(\varphi_n)\right) dx_{1:n}.$$

[2]Some authors use the terms *proposal density* or *instrumental density* interchangeably.

When the normalising constant is known analytically, we can calculate an unbiased importance sampling estimate — however, this generally has higher variance and this is not typically the case in the situations in which we are interested.

Furthermore, $I_n^{\mathrm{IS}}(\varphi)$ satisfies a Central Limit Theorem (CLT) with asymptotic variance

$$\frac{1}{N}\int \frac{\pi_n^2\left(x_{1:n}\right)}{q_n\left(x_{1:n}\right)}\left(\varphi_n\left(x_{1:n}\right)-I_n\left(\varphi_n\right)\right)^2 dx_{1:n}. \tag{27}$$

The bias being $\mathcal{O}(1/N)$ and the variance $\mathcal{O}(1/N)$, the mean-squared error given by the *squared* bias plus the variance is asymptotically dominated by the variance term.

For a given test function, $\varphi_n\left(x_{1:n}\right)$, it is easy to establish the importance distribution minimising the asymptotic variance of $I_n^{\mathrm{IS}}(\varphi_n)$. However, such a result is of minimal interest in a filtering context as this distribution depends on $\varphi_n\left(x_{1:n}\right)$ and we are typically interested in the expectations of several test functions. Moreover, even if we were interested in a single test function, say $\varphi_n\left(x_{1:n}\right)=x_n$, then selecting the optimal importance distribution at time n would have detrimental effects when we will try to obtain a sequential version of the algorithms (the optimal distribution for estimating $\varphi_{n-1}(x_{1:n-1})$ will almost certainly not be — even similar to — the marginal distribution of $x_{1:n-1}$ in the optimal distribution for estimating $\varphi_n(x_{1:n})$ and this will prove to be problematic).

A more appropriate approach in this context is to attempt to select the $q_n\left(x_{1:n}\right)$ which minimises the variance of the importance weights (or, equivalently, the variance of $\widehat{Z}_n$). Clearly, this variance is minimised for $q_n\left(x_{1:n}\right)=\pi_n\left(x_{1:n}\right)$. We cannot select $q_n\left(x_{1:n}\right)=\pi_n\left(x_{1:n}\right)$ as this is the reason we used IS in the first place. However, this simple result indicates that we should aim at selecting an IS distribution which is close as possible to the target. Also, although it is possible to construct samplers for which the variance is finite without satisfying this condition, it is advisable to select $q_n\left(x_{1:n}\right)$ so that $w_n\left(x_{1:n}\right)<C_n<\infty$.

3.3 Sequential Importance Sampling

We are now going to present an algorithm that admits a fixed computational complexity at each time step in important scenarios and thus addresses *Problem 2*. This solution involves selecting an importance distribution which has the following structure

$$\begin{aligned} q_n\left(x_{1:n}\right) &= q_{n-1}\left(x_{1:n-1}\right) q_n\left(x_n\middle|\, x_{1:n-1}\right) \\ &= q_1\left(x_1\right)\prod_{k=2}^{n} q_k\left(x_k\middle|\, x_{1:k-1}\right). \end{aligned} \tag{28}$$

Practically, this means that to obtain particles $X_{1:n}^i\sim q_n\left(x_{1:n}\right)$ at time n, we sample $X_1^i\sim q_1\left(x_1\right)$ at time 1 then $X_k^i\sim q_k\left(x_k\middle|\, X_{1:k-1}^i\right)$ at time k for $k=2,...,n$. The associated unnormalised weights can be computed recursively using the decomposition

$$\begin{aligned} w_n\left(x_{1:n}\right) &= \frac{\gamma_n\left(x_{1:n}\right)}{q_n\left(x_{1:n}\right)} \\ &= \frac{\gamma_{n-1}\left(x_{1:n-1}\right)}{q_{n-1}\left(x_{1:n-1}\right)}\frac{\gamma_n\left(x_{1:n}\right)}{\gamma_{n-1}\left(x_{1:n-1}\right) q_n\left(x_n\middle|\, x_{1:n-1}\right)} \end{aligned} \tag{29}$$

which can be written in the form

$$\begin{aligned} w_n\left(x_{1:n}\right) &= w_{n-1}\left(x_{1:n-1}\right)\cdot\alpha_n\left(x_{1:n}\right) \\ &= w_1\left(x_1\right)\prod_{k=2}^{n}\alpha_k\left(x_{1:k}\right) \end{aligned}$$

where the *incremental importance weight* function $\alpha_n(x_{1:n})$ is given by

$$\alpha_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})\, q_n(x_n|\, x_{1:n-1})}. \tag{30}$$

The SIS algorithm proceeds as follows, with each step carried out for $i = 1, \ldots, N$:

Sequential Importance Sampling

At time $n = 1$

- Sample $X_1^i \sim q_1(x_1)$.
- Compute the weights $w_1\left(X_1^i\right)$ and $W_1^i \propto w_1\left(X_1^i\right)$.

At time $n \geq 2$

- Sample $X_n^i \sim q_n(x_n|\, X_{1:n-1}^i)$.
- Compute the weights

$$w_n\left(X_{1:n}^i\right) = w_{n-1}\left(X_{1:n-1}^i\right) \cdot \alpha_n\left(X_{1:n}^i\right),$$
$$W_n^i \propto w_n\left(X_{1:n}^i\right).$$

At any time, n, we obtain the estimates $\widehat{\pi}_n(x_{1:n})$ (Eq. 23) and $\widehat{Z}_n$ (Eq. 24) of $\pi_n(x_{1:n})$ and Z_n, respectively. It is straightforward to check that a consistent estimate of Z_n/Z_{n-1} is also provided by the same set of samples:

$$\widehat{\frac{Z_n}{Z_{n-1}}} = \sum_{i=1}^{N} W_{n-1}^i \alpha_n\left(X_{1:n}^i\right).$$

This estimator is motivated by the fact that

$$\int \alpha_n(x_{1:n})\, \pi_{n-1}(x_{1:n-1})\, q_n(x_n|x_{1:n-1}) dx_{1:n} = \int \frac{\gamma_n(x_{1:n})\pi_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})}{\gamma_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})} dx_{1:n} = \frac{Z_n}{Z_{n-1}}.$$

In this sequential framework, it would seem that the only freedom the user has at time n is the choice of $q_n(x_n|\, x_{1:n-1})$[3]. A sensible strategy consists of selecting it so as to minimise the variance of $w_n(x_{1:n})$. It is straightforward to check that this is achieved by selecting

$$q_n^{\mathrm{opt}}(x_n|\, x_{1:n-1}) = \pi_n(x_n|\, x_{1:n-1})$$

as in this case the variance of $w_n(x_{1:n})$ conditional upon $x_{1:n-1}$ is zero and the associated incremental weight is given by

$$\alpha_n^{\mathrm{opt}}(x_{1:n}) = \frac{\gamma_n(x_{1:n-1})}{\gamma_{n-1}(x_{1:n-1})} = \frac{\int \gamma_n(x_{1:n})\, dx_n}{\gamma_{n-1}(x_{1:n-1})}.$$

Note that it is not always possible to sample from $\pi_n(x_n|\, x_{1:n-1})$. Nor is it always possible to compute $\alpha_n^{\mathrm{opt}}(x_{1:n})$. In these cases, one should employ an approximation of $q_n^{\mathrm{opt}}(x_n|\, x_{1:n-1})$ for $q_n(x_n|\, x_{1:n-1})$.

In those scenarios in which the time required to sample from $q_n(x_n|\, x_{1:n-1})$ and to compute $\alpha_n(x_{1:n})$ is independent of n (and this is, indeed, the case if q_n is chosen sensibly and one is concerned with a problem

[3]However, as we will see later, the key to many advanced SMC methods is the introduction of a sequence of target distributions which differ from the original target distributions.

such as filtering), it appears that we have provided a solution for *Problem 2*. However, it is important to be aware that the methodology presented here suffers from severe drawbacks. Even for standard IS, the variance of the resulting estimates increases exponentially with n (as is illustrated below; see also [28]). As SIS is nothing but a special version of IS in which we restrict ourselves to an importance distribution of the form (28) it suffers from the same problem. We demonstrate this using a very simple toy example.

Example. Consider the case where $\mathcal{X} = \mathbb{R}$ and

$$\begin{aligned} \pi_n(x_{1:n}) &= \prod_{k=1}^{n} \pi_n(x_k) = \prod_{k=1}^{n} \mathcal{N}(x_k; 0, 1), \\ \gamma_n(x_{1:n}) &= \prod_{k=1}^{n} \exp\left(-\frac{x_k^2}{2}\right), \\ Z_n &= (2\pi)^{n/2}. \end{aligned} \tag{31}$$

We select an importance distribution

$$q_n(x_{1:n}) = \prod_{k=1}^{n} q_k(x_k) = \prod_{k=1}^{n} \mathcal{N}\left(x_k; 0, \sigma^2\right).$$

In this case, we have $\mathbb{V}_{\mathrm{IS}}\left[\widehat{Z}_n\right] < \infty$ only for $\sigma^2 > \frac{1}{2}$ and

$$\frac{\mathbb{V}_{\mathrm{IS}}\left[\widehat{Z}_n\right]}{Z_n^2} = \frac{1}{N}\left[\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{n/2} - 1\right].$$

It can easily be checked that $\frac{\sigma^4}{2\sigma^2-1} > 1$ for any $\frac{1}{2} < \sigma^2 \neq 1$: the variance increases exponentially with n even in this simple case. For example, if we select $\sigma^2 = 1.2$ then we have a reasonably good importance distribution as $q_k(x_k) \approx \pi_n(x_k)$ but $N\frac{\mathbb{V}_{\mathrm{IS}}\left[\widehat{Z}_n\right]}{Z_n^2} \approx (1.103)^{n/2}$ which is approximately equal to 1.9×10^{21} for $n = 1000$! We would need to use $N \approx 2 \times 10^{23}$ particles to obtain a relative variance $\frac{\mathbb{V}_{\mathrm{IS}}\left[\widehat{Z}_n\right]}{Z_n^2} = 0.01$. This is clearly impracticable.

3.4 Resampling

We have seen that IS — and thus SIS — provides estimates whose variance increases, typically exponentially, with n. Resampling techniques are a key ingredient of SMC methods which (partially) solve this problem in some important scenarios.

Resampling is a very intuitive idea which has major practical and theoretical benefits. Consider first an IS approximation $\widehat{\pi}_n(x_{1:n})$ of the target distribution $\pi_n(x_{1:n})$. This approximation is based on weighted samples from $q_n(x_{1:n})$ and does not provide samples approximately distributed according to $\pi_n(x_{1:n})$. To obtain approximate samples from $\pi_n(x_{1:n})$, we can simply sample from its IS approximation $\widehat{\pi}_n(x_{1:n})$; that is we select $X_{1:n}^i$ with probability W_n^i. This operation is called resampling as it corresponds to sampling from an approximation $\widehat{\pi}_n(x_{1:n})$ which was itself obtained by sampling. If we are interested in obtaining N samples from $\widehat{\pi}_n(x_{1:n})$, then we can simply resample N times from $\widehat{\pi}_n(x_{1:n})$. This is equivalent to associating a number of offspring N_n^i with each particle $X_{1:n}^i$ in such a way that $N_n^{1:N} = \left(N_n^1, ..., N_n^N\right)$ follow a multinomial distribution with parameter vector $\left(N, W_n^{1:N}\right)$ and associating a weight of $1/N$ with each offspring. We approximate $\widehat{\pi}_n(x_{1:n})$ by the resampled empirical measure

$$\overline{\pi}_n(x_{1:n}) = \sum_{i=1}^{N} \frac{N_n^i}{N} \delta_{X_{1:n}^i}(x_{1:n}) \tag{32}$$

where $\mathbb{E}\left[N_n^i \middle| W_n^{1:N}\right] = NW_n^i$. Hence $\overline{\pi}_n(x_{1:n})$ is an unbiased approximation of $\widehat{\pi}_n(x_{1:n})$.

Improved unbiased resampling schemes have been proposed in the literature. These are methods of selecting N_n^i such that the unbiasedness property is preserved, and such that $\mathbb{V}\left[N_n^i \middle| W_n^{1:N}\right]$ is smaller than that obtained via the multinomial resampling scheme described above. To summarize, the three most popular algorithms found in the literature are, in descending order of popularity/efficiency:

Systematic Resampling Sample $U_1 \sim \mathcal{U}\left[0, \frac{1}{N}\right]$ and define $U_i = U_1 + \frac{i-1}{N}$ for $i = 2, ..., N$, then set $N_n^i = \left|\left\{U_j : \sum_{k=1}^{i-1} W_n^k \leq U_j \leq \sum_{k=1}^{i} W_n^k\right\}\right|$ with the convention $\sum_{k=1}^{0} := 0$. It is straightforward to establish that this approach is unbiased.

Residual Resampling Set $\widetilde{N}_n^i = \lfloor N W_n^i \rfloor$, sample $\overline{N}_n^{1:N}$ from a multinomial of parameters $\left(N, \overline{W}_n^{1:N}\right)$ where $\overline{W}_n^i \propto W_n^i - N^{-1}\widetilde{N}_n^i$ then set $N_n^i = \widetilde{N}_n^i + \overline{N}_n^i$. This is very closely related to breaking the empirical CDF up into N components and then sampling once from each of those components: the stratified resampling approach of [7].

Multinomial Resampling Sample $N_n^{1:N}$ from a multinomial of parameters $\left(N, W_n^{1:N}\right)$.

Note that it is possible to sample efficiently from a multinomial distribution in $\mathcal{O}(N)$ operations. However, the systematic resampling algorithm introduced in [25] is the most widely-used algorithm in the literature as it is extremely easy to implement and outperforms other resampling schemes in most scenarios (although this is not guaranteed in general [13]).

Resampling allows us to obtain samples distributed approximately according to $\pi_n(x_{1:n})$, but it should be clear that if we are interested in estimating $I_n(\varphi_n)$ then we will obtain an estimate with lower variance using $\widehat{\pi}_n(x_{1:n})$ than that which we would have obtained by using $\overline{\pi}_n(x_{1:n})$. By resampling we indeed add some extra "noise"— as shown by [9]. However, an important advantage of resampling is that it allows us to remove of particles with low weights with a high probability. In the sequential framework in which we are interested, this is extremely useful as we do not want to carry forward particles with low weights and we want to focus our computational efforts on regions of high probability mass. Clearly, there is always the possibility than a particle having a low weight at time n could have an high weight at time $n+1$, in which case resampling could be wasteful. It is straightforward to consider artificial problems for which this is the case. However, we will show that in the estimation problems we are looking at the resampling step is provably beneficial. Intuitively, resampling can be seen to provide stability in the future at the cost of an increase in the immediate Monte Carlo variance. This concept will be made more precise in section 3.6.

3.5 A Generic Sequential Monte Carlo Algorithm

SMC methods are a combination of SIS and resampling. At time 1, we compute the IS approximation $\widehat{\pi}_1(x_1)$ of $\pi_1(x_1)$ which is a weighted collection of particles $\left\{W_1^i, X_1^i\right\}$. Then we use a resampling step to eliminate (with high probability) those particles with low weights and multiply those with high weights. We denote by $\left\{\frac{1}{N}, \overline{X}_1^i\right\}$ the collection of equally-weighted resampled particles. Remember that each original particle X_1^i has N_1^i offspring so there exist N_1^i distinct indexes $j_1 \neq j_2 \neq \cdots \neq j_{N_1^i}$ such that $\overline{X}_1^{j_1} = \overline{X}_1^{j_2} = \cdots = \overline{X}_1^{j_{N_1^i}} = X_1^i$. After the resampling step, we follow the SIS strategy and sample $X_2^i \sim q_2(x_2|\,\overline{X}_1^i)$. Thus $\left(\overline{X}_1^i, X_2^i\right)$ is approximately distributed according to $\pi_1(x_1)\,q_2(x_2|\,x_1)$. Hence the corresponding importance weights in this case are simply equal to the incremental weights $\alpha_2(x_{1:2})$. We then resample the particles with respect to these normalised weights and so on. To summarise, the algorithm proceeds as follows (this algorithm is sometimes referred to as Sequential Importance Resampling (SIR) or Sequential Importance Sampling and Resampling (SIS/R)).

Sequential Monte Carlo

At time $n = 1$

- Sample $X_1^i \sim q_1(x_1)$.
- Compute the weights $w_1\left(X_1^i\right)$ and $W_1^i \propto w_1\left(X_1^i\right)$.
- Resample $\left\{W_1^i, X_1^i\right\}$ to obtain N equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_1^i\right\}$.

At time $n \geq 2$

- Sample $X_n^i \sim q_n(x_n|\, \overline{X}_{1:n-1}^i)$ and set $X_{1:n}^i \leftarrow \left(\overline{X}_{1:n-1}^i, X_n^i\right)$.
- Compute the weights $\alpha_n\left(X_{1:n}^i\right)$ and $W_n^i \propto \alpha_n\left(X_{1:n}^i\right)$.
- Resample $\left\{W_n^i, X_{1:n}^i\right\}$ to obtain N new equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_{1:n}^i\right\}$.

At any time n, this algorithm provides two approximations of $\pi_n\left(x_{1:n}\right)$. We obtain

$$\widehat{\pi}_n\left(x_{1:n}\right) = \sum_{i=1}^{N} W_n^i \delta_{X_{1:n}^i}\left(x_{1:n}\right) \tag{33}$$

after the sampling step and

$$\overline{\pi}_n\left(x_{1:n}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\overline{X}_{1:n}^i}\left(x_{1:n}\right) \tag{34}$$

after the resampling step. The approximation (33) is to be preferred to (34). We also obtain an approximation of Z_n/Z_{n-1} through

$$\widehat{\frac{Z_n}{Z_{n-1}}} = \frac{1}{N}\sum_{i=1}^{N} \alpha_n\left(X_{1:n}^i\right).$$

As we have already mentioned, resampling has the effect of removing particles with low weights and multiplying particles with high weights. However, this is at the cost of immediately introducing some additional variance. If particles have unnormalised weights with a small variance then the resampling step might be unnecessary. Consequently, in practice, it is more sensible to resample only when the variance of the unnormalised weights is superior to a pre-specified threshold. This is often assessed by looking at the variability of the weights using the so-called Effective Sample Size (ESS) criterion [30, pp. 35-36], which is given (at time n) by

$$ESS = \left(\sum_{i=1}^{N}\left(W_n^i\right)^2\right)^{-1}.$$

Its interpretation is that in a simple IS setting, inference based on the N weighted samples is approximately equivalent (in terms of estimator variance) to inference based on ESS perfect samples from the target distribution. The ESS takes values between 1 and N and we resample only when it is below a threshold N_T; typically $N_T = N/2$. Alternative criteria can be used such as the entropy of the weights $\left\{W_n^i\right\}$ which achieves its maximum value when $W_n^i = \frac{1}{N}$. In this case, we resample when the entropy is below a given threshold.

Sequential Monte Carlo with Adaptive Resampling

At time $n = 1$

• Sample $X_1^i \sim q_1(x_1)$.

• Compute the weights $w_1\left(X_1^i\right)$ and $W_1^i \propto w_1\left(X_1^i\right)$.

• If resampling criterion satisfied then resample $\left\{W_1^i, X_1^i\right\}$ to obtain N equally weighted particles $\left\{\frac{1}{N}, \overline{X}_1^i\right\}$ and set $\left\{\overline{W}_1^i, \overline{X}_1^i\right\} \leftarrow \left\{\frac{1}{N}, \overline{X}_1^i\right\}$, otherwise set $\left\{\overline{W}_1^i, \overline{X}_1^i\right\} \leftarrow \left\{W_1^i, X_1^i\right\}$.

At time $n \geq 2$

• Sample $X_n^i \sim q_n(x_n | \overline{X}_{1:n-1}^i)$ and set $X_{1:n}^i \leftarrow \left(\overline{X}_{1:n-1}^i, X_n^i\right)$.

• Compute the weights $\alpha_n\left(X_{1:n}^i\right)$ and $W_n^i \propto \overline{W}_{n-1}^i \alpha_n\left(X_{1:n}^i\right)$.

• If resampling criterion satisfied, then resample $\left\{W_n^i, X_{1:n}^i\right\}$ to obtain N equally weighted particles $\left\{\frac{1}{N}, \overline{X}_{1:n}^i\right\}$ and set $\left\{\overline{W}_n^i, \overline{X}_n^i\right\} \leftarrow \left\{\frac{1}{N}, \overline{X}_n^i\right\}$, otherwise set $\left\{\overline{W}_n^i, \overline{X}_n^i\right\} \leftarrow \left\{W_n^i, X_n^i\right\}$.

In this context too we have two approximations of $\pi_n\left(x_{1:n}\right)$

$$\widehat{\pi}_n\left(x_{1:n}\right) = \sum_{i=1}^{N} W_n^i \delta_{X_{1:n}^i}\left(x_{1:n}\right), \tag{35}$$

$$\overline{\pi}_n\left(x_{1:n}\right) = \sum_{i=1}^{N} \overline{W}_n^i \delta_{\overline{X}_{1:n}^i}\left(x_{1:n}\right)$$

which are equal if no resampling step is used at time n. We may also estimate Z_n/Z_{n-1} through

$$\widehat{\frac{Z_n}{Z_{n-1}}} = \sum_{i=1}^{N} \overline{W}_{n-1}^i \alpha_n\left(X_{1:n}^i\right). \tag{36}$$

SMC methods involve systems of particles which interact (via the resampling mechanism) and, consequently, obtaining convergence results is a much more difficult task than it is for SIS where standard results (iid asymptotics) apply. However, there are numerous sharp convergence results available for SMC; see [10] for an introduction to the subject and the monograph of Del Moral [11] for a complete treatment of the subject. An explicit treatment of the case in which resampling is performed adaptively is provided by [12].

The presence or absence of degeneracy is the factor which most often determines whether an SMC algorithm works in practice. However strong the convergence results available for limitingly large samples may be, we cannot expect good performance if the *finite* sample which is actually used is degenerate. Indeed, some degree of degeneracy is inevitable in all but trivial cases: if SMC algorithms are used for sufficiently many time steps every resampling step reduces the number of unique values representing X_1, for example. For this reason, any SMC algorithm which relies upon the distribution of full paths $x_{1:n}$ will fail for large enough n for any finite sample size, N, in spite of the asymptotic justification. It is intuitive that one should endeavour to employ algorithms which do not depend upon the full path of the samples, but only upon the distribution of some finite component $x_{n-L:n}$ for some fixed L which is independent of n. Furthermore, ergodicity (a tendency for the future to be essentially independent of the distant past) of the underlying system will prevent the accumulation of errors over time. These concepts are precisely characterised by existing convergence results, some of the most important of which are summarised and interpreted in section 3.6.

Although sample degeneracy emerges as a consequence of resampling, it is really a manifestation of a deeper problem — one which resampling actually mitigates. It is inherently impossible to accurately represent a distribution on a space of arbitrarily high dimension with a sample of fixed, finite size. Sample impoverishment is a term which is often used to describe the situation in which very few different particles have significant weight. This problem has much the same effect as sample degeneracy and occurs, in the absence of resampling, as the inevitable consequence of multiplying together incremental importance weights from a large number of time steps. It is, of course, not possible to circumvent either problem by increasing the number of samples at every iteration to maintain a constant effective sample size as this would lead to an exponential growth in the number of samples required. This sheds some light on the resampling mechanism: it "resets the system" in such a way that its representation of final time marginals remains well behaved at the expense of further diminishing the quality of the path-samples. By focusing on the fixed-dimensional final time marginals in this way, it allows us to circumvent the problem of increasing dimensionality.

3.6 Convergence Results for Sequential Monte Carlo Methods

Here, we briefly discuss selected convergence results for SMC. We focus on the CLT as it allows us to clearly understand the benefits of the resampling step and why it "works". If multinomial resampling is used at every iteration[4], then the associated SMC estimates of $\widehat{Z}_n/Z_n$ and $I_n(\varphi_n)$ satisfy a CLT and their respective asymptotic variances are given by

$$\frac{1}{N}\left(\int \frac{\pi_n^2(x_1)}{q_1(x_1)}dx_1 - 1 + \sum_{k=2}^{n}\int \frac{\pi_n^2(x_{1:k})}{\pi_{k-1}(x_{1:k-1})\,q_k(x_k|x_{1:k-1})}dx_{k-1:k} - 1\right) \tag{37}$$

and

$$\begin{aligned}&\int \tfrac{\pi_1^2(x_1)}{q_1(x_1)}\left(\int \varphi_n(x_{1:n})\,\pi_n(x_{2:n}|x_1)\,dx_{2:n} - I_n(\varphi_n)\right)^2 dx_1 \\ &+\textstyle\sum_{k=2}^{n-1}\int \frac{\pi_n^2(x_{1:k})}{\pi_{k-1}(x_{1:k-1})q_k(x_k|x_{1:k-1})}\left(\int \varphi_n(x_{1:n})\,\pi_n(x_{k+1:n}|x_{1:k})\,dx_{k+1:n} - I_n(\varphi_n)\right)^2 dx_{1:k} \\ &+\int \frac{\pi_n^2(x_{1:n})}{\pi_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})}\left(\varphi_n(x_{1:n}) - I_n(\varphi_n)\right)^2 dx_{1:n}.\end{aligned} \tag{38}$$

A short and elegant proof of this result is given in [11, Chapter 9]; see also [9]. These expression are very informative. They show that the resampling step has the effect of "resetting" the particle system whenever it is applied. Comparing (26) to (37), we see that the SMC variance expression has replaced the importance distribution $q_n(x_{1:n})$ in the SIS variance with the importance distributions $\pi_{k-1}(x_{1:k-1})\,q_k(x_k|x_{1:k-1})$ obtained after the resampling step at time $k-1$. Moreover, we will show that in important scenarios the variances of SMC estimates are orders of magnitude smaller than the variances of SIS estimates.

Let us first revisit the toy example discussed in section 3.3.

Example (continued). In this case, it follows from (37) that the asymptotic variance is finite only when $\sigma^2 > \frac{1}{2}$ and

$$\begin{aligned}\frac{\mathbb{V}_{\mathrm{SMC}}\left[\widehat{Z}_n\right]}{Z_n^2} &\approx \frac{1}{N}\left[\int \frac{\pi_n^2(x_1)}{q_1(x_1)}dx_1 - 1 + \sum_{k=2}^{n}\int \frac{\pi_n^2(x_k)}{q_k(x_k)}dx_k - 1\right] \\ &= \frac{n}{N}\left[\left(\frac{\sigma^4}{2\sigma^2-1}\right)^{1/2} - 1\right]\end{aligned}$$

[4]Similar expressions can be established when a lower variance resampling strategy such as residual resampling is used and when resampling is performed adaptively [12]. The results presented here are sufficient to guide the design of particular algorithms and the additional complexity involved in considering more general scenarios serves largely to produce substantially more complex expressions which obscure the important points.

compared to

$$\frac{\mathbb{V}_{\mathrm{IS}}\left[\widehat{Z}_n\right]}{Z_n^2} = \frac{1}{N}\left[\left(\frac{\sigma^4}{2\sigma^2-1}\right)^{n/2} - 1\right].$$

The asymptotic variance of the SMC estimate increases linearly with n in contrast to the exponential growth of the IS variance. For example, if we select $\sigma^2 = 1.2$ then we have a reasonably good importance distribution as $q_k(x_k) \approx \pi_n(x_k)$. In this case, we saw that it is necessary to employ $N \approx 2 \times 10^{23}$ particles in order to obtain $\frac{\mathbb{V}_{\mathrm{IS}}[\widehat{Z}_n]}{Z_n^2} = 10^{-2}$ for $n = 1000$. Whereas to obtain the same performance, $\frac{\mathbb{V}_{\mathrm{SMC}}[\widehat{Z}_n]}{Z_n^2} = 10^{-2}$, SMC requires the use of just $N \approx 10^4$ particles: an improvement by 19 orders of magnitude.

This scenario is overly favourable to SMC as the target (31) factorises. However, generally speaking, the major advantage of SMC over IS is that it allows us to exploit the forgetting properties of the model under study as illustrated by the following example.

Example. Consider the following more realistic scenario where

$$\gamma_n(x_{1:n}) = \mu(x_1)\prod_{k=2}^{n} M_k(x_k|x_{k-1})\prod_{k=1}^{n} G_k(x_k)$$

with μ a probability distribution, M_k a Markov transition kernel and G_k a positive "potential" function. Essentially, filtering corresponds to this model with $M_k(x_k|x_{k-1}) = f(x_k|x_{k-1})$ and the time inhomogeneous potential function $G_k(x_k) = g(y_k|x_k)$. In this case, $\pi_k(x_k|x_{1:k-1}) = \pi_k(x_k|x_{k-1})$ and we would typically select an importance distribution $q_k(x_k|x_{1:k-1})$ with the same Markov property $q_k(x_k|x_{1:k-1}) = q_k(x_k|x_{k-1})$. It follows that (37) is equal to

$$\frac{1}{N}\left(\int \frac{\pi_1^2(x_1)}{q_1(x_1)}dx_1 - 1 + \sum_{k=2}^{n}\int \frac{\pi_n^2(x_{k-1:k})}{\pi_{k-1}(x_{k-1})\,q_k(x_k|x_{k-1})}dx_{k-1:k} - 1\right)$$

and (38), for $\varphi_n(x_{1:n}) = \varphi(x_n)$, equals:

$$\begin{array}{l}\int \frac{\pi_1^2(x_1)}{q_1(x_1)}\left(\int \varphi(x_n)\,\pi_n(x_n|x_1)\,dx_{2:n} - I_n(\varphi)\right)^2 dx_1 \\ + \sum_{k=2}^{n-1}\int \frac{\pi_n^2(x_{k-1:k})}{\pi_{k-1}(x_{k-1})q_k(x_k|x_{k-1})}\left(\int \varphi(x_n)\,\pi_n(x_n|x_k)\,dx_k - I_n(\varphi)\right)^2 dx_{k-1:k} \\ + \int \frac{\pi_n^2(x_{n-1:n})}{\pi_{n-1}(x_{n-1})q_n(x_n|x_{n-1})}\left(\varphi(x_n) - I_n(\varphi)\right)^2 dx_{n-1:n},\end{array}$$

where we use the notation $I_n(\varphi)$ for $I_n(\varphi_n)$. In many realistic scenarios, the model associated with $\pi_n(x_{1:n})$ has some sort of ergodic properties; i.e. $\forall x_k, x_k' \in \mathcal{X}$ $\pi_n(x_n|x_k) \approx \pi_n(x_n|x_k')$ for large enough $n-k$. In layman's terms, at time n what happened at time k is irrelevant if $n-k$ is large enough. Moreover, this often happens exponentially fast; that is for any (x_k, x_k')

$$\frac{1}{2}\int |\pi_n(x_n|x_k) - \pi_n(x_n|x_k')|\,dx_n \leq \beta^{n-k}$$

for some $\beta < 1$. This property can be used to establish that for bounded functions $\varphi \leq \|\varphi\|$

$$\left|\int \varphi(x_n)\,\pi_n(x_n|x_k)\,dx_n - I(\varphi)\right| \leq \beta^{n-k}\|\varphi\|$$

and under weak additional assumptions we have

$$\frac{\pi_n^2(x_{k-1:k})}{\pi_{k-1}(x_{k-1})\,q_k(x_k|x_{k-1})} \leq A$$

for a finite constant A. Hence it follows that

$$\frac{\mathbb{V}_{\mathrm{SMC}}\left[\widehat{Z}_n\right]}{Z_n^2} \leq \frac{C\cdot n}{N},$$

$$\mathbb{V}_{\mathrm{SMC}}\left[\widehat{I}_n(\varphi)\right] \leq \frac{D}{N}.$$

for some finite constants C, D that are independent of n. These constants typically increase polynomially/exponentially with the dimension of the state-space $\mathcal{X}$ and decrease as $\beta \to 0$.

3.7 Summary

We have presented a generic SMC algorithm which approximates $\{\pi_n(x_{1:n})\}$ and $\{Z_n\}$ sequentially in time.

- Wherever it is possible to sample from $q_n(x_n|x_{1:n-1})$ and evaluate $\alpha_n(x_{1:n})$ in a time independent of n, this leads to an algorithm whose computational complexity does not increase with n.
- For any k, there exists $n > k$ such that the SMC approximation of $\pi_n(x_{1:k})$ consists of a single particle because of the successive resampling steps. It is thus impossible to get a "good" SMC approximation of the joint distributions $\{\pi_n(x_{1:n})\}$ when n is too large. This can easily be seen in practice, by monitoring the number of distinct particles approximating $\pi_n(x_1)$.
- However, under mixing conditions, this SMC algorithm is able to provide estimates of marginal distributions of the form $\pi_n(x_{n-L+1:n})$ and estimates of Z_n/Z_{n-1} whose variance is uniformly bounded with n. This property is crucial and explains why SMC methods "work" in many realistic scenarios.
- Practically, one should keep in mind that the variance of SMC estimates can only expected to be reasonable if the variance of the incremental weights is small. In particular, this requires that we can only expect to obtain good performance if $\pi_n(x_{1:n-1}) \approx \pi_{n-1}(x_{1:n-1})$ and $q_n(x_n|x_{1:n-1}) \approx \pi_n(x_n|x_{1:n-1})$; that is if the successive distributions we want to approximate do not differ much one from each other and the importance distribution is a reasonable approximation of the "optimal" importance distribution. However, if successive distributions differ significantly, it is often possible to design an artificial sequence of distributions to "bridge" this transition [21, 31].

4 Particle Filtering

Remember that in the filtering context, we want to be able to compute a numerical approximation of the distribution $\{p(x_{1:n}|y_{1:n})\}_{n\geq 1}$ *sequentially* in time. A direct application of the SMC methods described earlier to the sequence of target distributions $\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n})$ yields a popular class of particle filters. More elaborate sequences of target and proposal distributions yield various more advanced algorithms. For ease of presentation, we present algorithms in which we resample at each time step. However, in practice we recommend only resampling when the ESS is below a threshold and employing the systematic resampling scheme.

4.1 SMC for Filtering

First, consider the simplest case in which $\gamma_n(x_{1:n}) = p(x_{1:n}, y_{1:n})$ is chosen, yielding $\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n})$ and $Z_n = p(y_{1:n})$. Practically, it is only necessary to select the importance distribution $q_n(x_n|x_{1:n-1})$. We have seen that in order to minimise the variance of the importance weights at time n, we should select $q_n^{\mathrm{opt}}(x_n|x_{1:n-1}) = \pi_n(x_n|x_{1:n-1})$ where

$$\begin{aligned}\pi_n(x_n|x_{1:n-1}) &= p(x_n|y_n, x_{n-1}) \\ &= \frac{g(y_n|x_n) f(x_n|x_{n-1})}{p(y_n|x_{n-1})},\end{aligned} \tag{39}$$

and the associated incremental importance weight is $\alpha_n(x_{1:n}) = p(y_n|x_{n-1})$. In many scenarios, it is not possible to sample from this distribution but we should aim to approximate it. In any case, it shows that we should use an importance distribution of the form

$$q_n(x_n|x_{1:n-1}) = q(x_n|y_n, x_{n-1}) \tag{40}$$

and that there is nothing to be gained from building importance distributions depending also upon $(y_{1:n-1}, x_{1:n-2})$ — although, at least in principle, in some settings there may be advantages to using information from subsequent observations if they are available. Combining (29), (30) and (40), the incremental weight is given by

$$\alpha_n(x_{1:n}) = \alpha_n(x_{n-1:n}) = \frac{g(y_n|x_n)f(x_n|x_{n-1})}{q(x_n|y_n, x_{n-1})}.$$

The algorithm can thus be summarised as follows.

SIR/SMC for Filtering

At time $n = 1$

- Sample $X_1^i \sim q(x_1|y_1)$.
- Compute the weights $w_1(X_1^i) = \frac{\mu(X_1^i)g(y_1|X_1^i)}{q(X_1^i|y_1)}$ and $W_1^i \propto w_1(X_1^i)$.
- Resample $\{W_1^i, X_1^i\}$ to obtain N equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_1^i\right\}$.

At time $n \geq 2$

- Sample $X_n^i \sim q(x_n|y_n, \overline{X}_{n-1}^i)$ and set $X_{1:n}^i \leftarrow \left(\overline{X}_{1:n-1}^i, X_n^i\right)$.
- Compute the weights $\alpha_n(X_{n-1:n}^i) = \frac{g(y_n|X_n^i)f(X_n^i|X_{n-1}^i)}{q(X_n^i|y_n, X_{n-1}^i)}$ and $W_n^i \propto \alpha_n(X_{n-1:n}^i)$.
- Resample $\{W_n^i, X_{1:n}^i\}$ to obtain N new equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_{1:n}^i\right\}$.

We obtain at time n

$$\widehat{p}(x_{1:n}|y_{1:n}) = \sum_{i=1}^{N} W_n^i \delta_{X_{1:n}^i}(x_{1:n}),$$

$$\widehat{p}(y_n|y_{1:n-1}) = \sum_{i=1}^{N} W_{n-1}^i \alpha_n(X_{n-1:n}^i).$$

However, if we are interested only in approximating the marginal distributions $\{p(x_n|y_{1:n})\}$ and $\{p(y_{1:n})\}$ then we need to store only the terminal-value particles $\{X_{n-1:n}^i\}$ to be able to compute the weights: the algorithm's storage requirements do not increase over time.

Many techniques have been proposed to design "efficient" importance distributions $q(x_n|y_n, x_{n-1})$ which approximate $p(x_n|y_n, x_{n-1})$. In particular the use of standard suboptimal filtering techniques such as the Extended Kalman Filter or the Unscented Kalman Filter to obtain importance distributions is very popular in the literature [14, 37]. The use of local optimisation techniques to design $q(x_n|y_n, x_{n-1})$ centred around the mode of $p(x_n|y_n, x_{n-1})$ has also been advocated [33, 34].

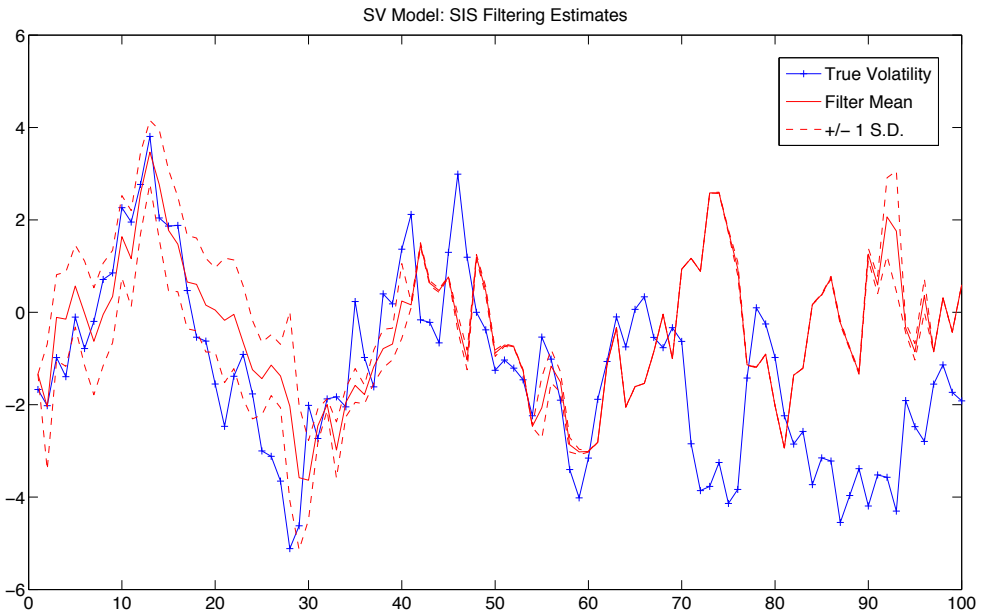

Figure 2: Filtering estimates obtained for the stochastic volatility model using SIS. At each time the mean and standard deviation of x_n conditional upon $y_{1:n}$ is estimated using the particle set. It initially performs reasonably, but the approximation eventually collapses: the estimated mean begins to diverge from the truth and the estimate of the standard deviation is inaccurately low.

4.1.1 Example: Stochastic Volatility

Returning to example 4 and the simulated data shown in figure 1, we are able to illustrate the performance of SMC algorithms with and without resampling steps in a filtering context.

An SIS algorithm (corresponding to the above SMC algorithm without a resampling step) with $N = 1000$ particles, in which the conditional prior is employed as a proposal distribution (leading to an algorithm in which the particle weights are proportional to the likelihood function) produces the output shown in figure 2. Specifically, at each iteration, n, of the algorithm the conditional expectation and standard deviation of x_n, given $y_{1:n}$ is obtained. It can be seen that the performance is initially good, but after a few iterations the estimate of the mean becomes inaccurate, and the estimated standard deviation shrinks to a very small value. This standard deviation is an estimate of the standard deviation of the conditional posterior obtained via the particle filter: it is not a measure of the standard deviation of the estimator. Such an estimate can be obtained by considering several independent particle filters run on the same data and would illustrate the high variability of estimates obtained by a poorly-designed algorithm such as this one. In practice, approximations such as the effective sample size are often used as surrogates to characterise the uncertainty of the filter estimates but these perform well only if the filter is providing a reasonable approximation of the conditional distributions of interest. Figure 3 supports the theory that the failure of the algorithm after a few iterations is due to weight degeneracy, showing that the number of particles with significant weight falls rapidly.

The SIR algorithm described above was also applied to this problem with the same proposal distribution and number of particles as were employed in the SIS case. For simplicity, multinomial resampling was applied at every iteration. Qualitatively, the same features would be observed if a more sophisticated algorithm were employed, or adaptive resampling were used although these approaches would lessen the severity of the path-degeneracy problem. Figure 4 shows the distribution of particle weights for this algorithm. Notice that unlike the SIS algorithm shown previously, there are many particles with

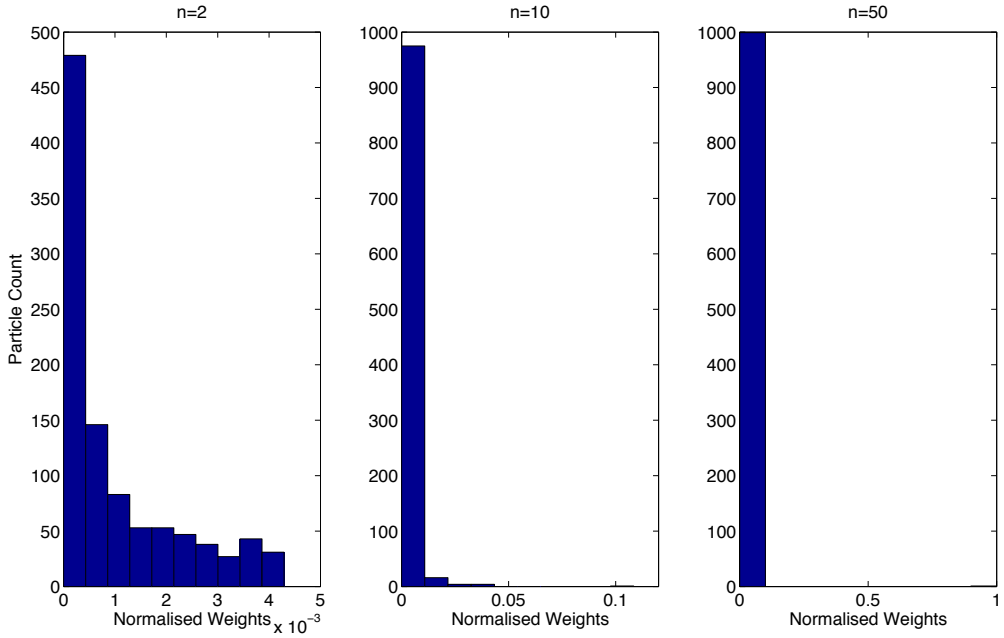

Figure 3: Empirical distributions of the particle weights obtained with the SIS algorithm for the stochastic volatility model at iterations 2, 10 and 50. Although the algorithm is reasonably initialised, by iteration 10 only a few tens of particles have significant weight and by iteration 50 a single particle is dominant.

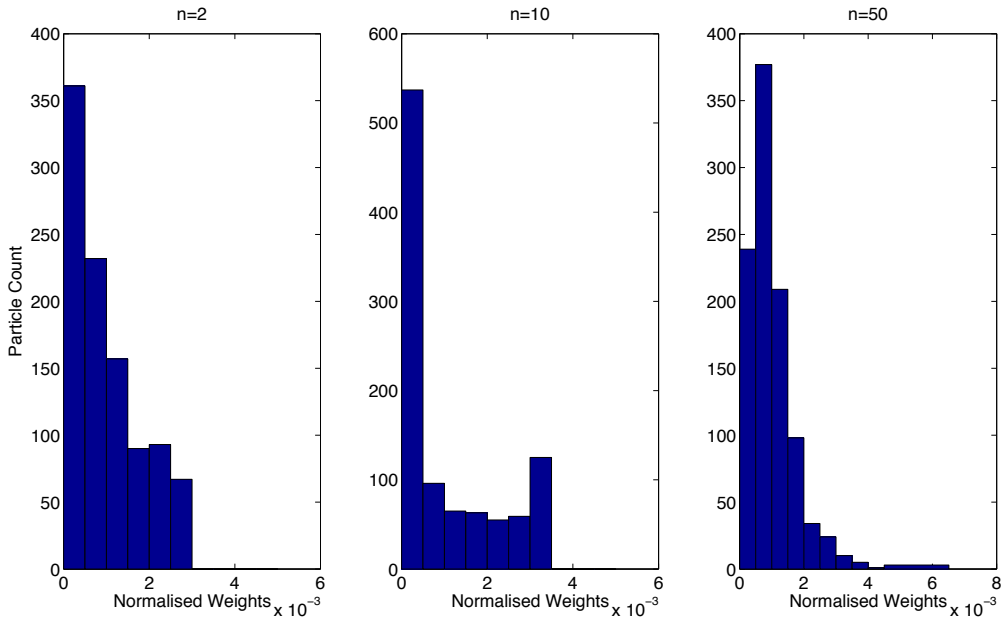

Figure 4: Empirical distribution of particle weights for an SIR algorithm applied to the stochastic volatility model. Notice that there is no evidence of weight degeneracy in contrast to the SIS case. This comes at the cost of reducing the quality of the path-space representation.

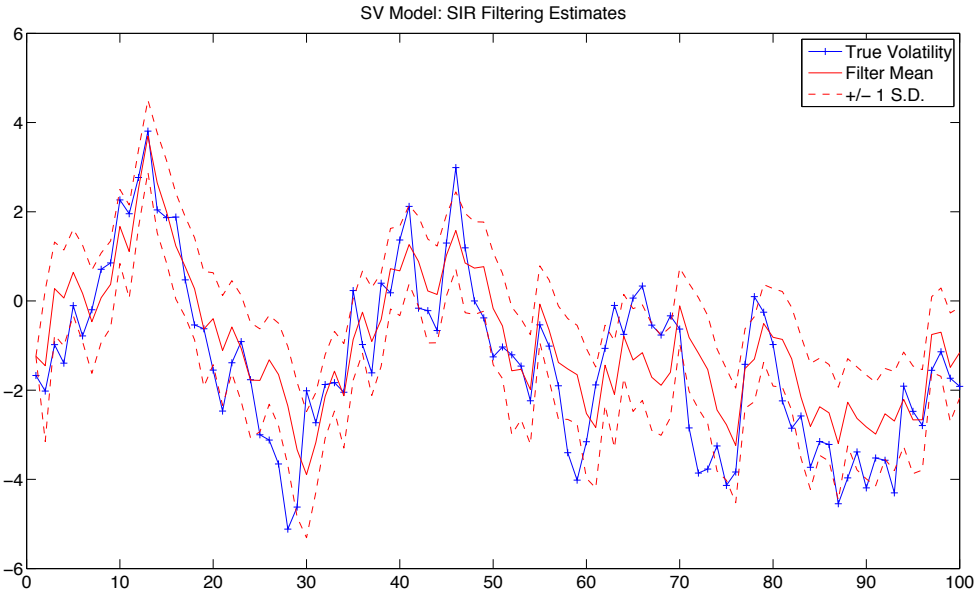

Figure 5: SIR Filtering estimates for the SV model.

significant weight at all three time points. It is important to note that while this is encouraging it is not evidence that the algorithm is performing well: it provides no information about the path-space distribution and, in fact, it is easy to construct poorly-performing algorithms which appear to have a good distribution of particle weights (for instance, consider a scenario in which the target is relatively flat in its tails but sharply concentrated about a mode; if the proposal has very little mass in the vicinity of the mode then it is likely that a collection of very similar importance weights will be obtained — but the sample thus obtained does not characterise the distribution of interest well). Figure 5 shows that the algorithm does indeed produce a reasonable estimate and plausible credible interval. And, as we expect, a problem does arise when we consider the smoothing distributions $p(x_n|y_{1:500})$ as shown in figure 6: the estimate and credible interval is unreliable for $n \ll 500$. This is due to the degeneracy caused at the beginning of the path by repeated resampling. In contrast the smoothed estimate for $n \approx 500$ (not shown) is reasonable.

4.2 Auxiliary Particle Filtering

As was discussed above, the optimal proposal distribution (in the sense of minimising the variance of importance weights) when performing standard particle filtering is $q(x_n|y_n, x_{n-1}) = p(x_n|y_n, x_{n-1})$. Indeed, $\alpha_n(x_{n-1:n})$ is independent of x_n in this case so it is possible to interchange the order of the sampling and resampling steps. Intuitively, this yields a better approximation of the distribution as it provides a greater number of distinct particles to approximate the target. This is an example of a general principle: resampling, if it is to be applied in a particular iteration, should be performed before, rather than after, any operation that doesn't influence the importance weights in order to minimise the loss of information.

It is clear that if importance weights are independent of the new state and the proposal distribution corresponds to the marginal distribution of the proposed states then weighting, resampling and then sampling corresponds to a reweighting to correct for the discrepancy between the old and new marginal distribution of the earlier states, resampling to produce an unweighted sample and then generation of the new state from its conditional distribution. This intuition can easily be formalised.

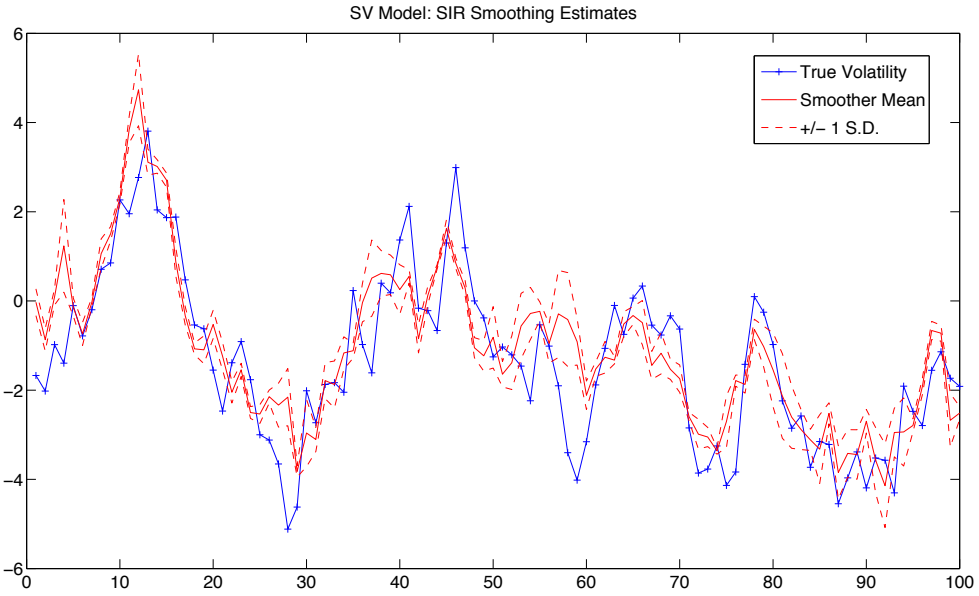

Figure 6: SIR Smoothing estimates for the SV model.

However, in general, the incremental importance weights do depend upon the new states and this straightforward change of order becomes impossible. In a sense, this interchange of sampling and resampling produces an algorithm in which information from the next observation is used to determine which particles should survive resampling at a given time (to see this, consider weighting and resampling occurring as the very last step of the iteration before the current one, rather than as the first step of that iteration). It is desirable to find methods for making use of this future information in a more general setting, so that we can obtain the same advantage in situations in which it is not possible to make use of the optimal proposal distribution.

The Auxiliary Particle Filter (APF) is an alternative algorithm which does essentially this. It was originally introduced in [33] using auxiliary variables — hence its name. Several improvements were proposed to reduce its variance [7, 34]. We present here the version of the APF presented in [7] which only includes one resampling step at each time instance. It has long been realised that, experimentally, this version outperforms the original two stage resampling algorithm proposed in [33] and is widely used; see [7] for a comparison of both approaches. The APF is a look ahead method where at time n we try to predict which samples will be in regions of high probability masses at time $n+1$.

It was shown in [23] that the APF can be reinterpreted as a *standard* SMC algorithm applied to the following sequence of target distributions

$$\gamma_n\left(x_{1:n}\right) = p\left(x_{1:n}, y_{1:n}\right)\widetilde{p}\left(y_{n+1}\middle| x_n\right) \tag{41}$$

with $\widetilde{p}\left(y_{n+1}\middle| x_n\right)$ chosen as an approximation of the predictive likelihood $p\left(y_{n+1}\middle| x_n\right)$ if it is not known analytically. It follows that $\pi_n\left(x_{1:n}\right)$ is an approximation of $p\left(x_{1:n}\middle| y_{1:n+1}\right)$ denoted $\widetilde{p}\left(x_{1:n}\middle| y_{1:n+1}\right)$ given by

$$\pi_n\left(x_{1:n}\right) = \widetilde{p}\left(x_{1:n}\middle| y_{1:n+1}\right) \propto p\left(x_{1:n}\middle| y_{1:n}\right)\widetilde{p}\left(y_{n+1}\middle| x_n\right) \tag{42}$$

In the APF we also use an importance distribution $q_n\left(x_n\middle| x_{1:n-1}\right)$ of the form (40) which is typically an approximation of (39). Note that (39) is different from $\pi_n\left(x_n\middle| x_{1:n-1}\right)$ in this scenario. Even if we could sample from $\pi_n\left(x_n\middle| x_{1:n-1}\right)$, one should remember that in this case the object of inference is not

$\pi_n(x_{1:n}) = \widetilde{p}(x_{1:n}|y_{1:n+1})$ but $p(x_{1:n}|y_{1:n})$. The associated incremental weight is given by

$$\begin{aligned}\alpha_n(x_{n-1:n}) &= \frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})\, q_n(x_n|x_{1:n-1})} \\ &= \frac{g(y_n|x_n)\, f(x_n|x_{n-1})\, \widetilde{p}(y_{n+1}|x_n)}{\widetilde{p}(y_n|x_{n-1})\, q(x_n|y_n, x_{n-1})}.\end{aligned}$$

To summarize, the APF proceeds as follows.

Auxiliary Particle Filtering

At time $n = 1$

- Sample $X_1^i \sim q(x_1|y_1)$.
- Compute the weights $w_1\left(X_1^i\right) = \frac{\mu\left(X_1^i\right) g\left(y_1|X_1^i\right) \widetilde{p}\left(y_2|X_1^i\right)}{q\left(X_1^i|y_1\right)}$ and $W_1^i \propto w_1\left(X_1^i\right)$.
- Resample $\left\{W_1^i, X_1^i\right\}$ to obtain N equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_1^i\right\}$.

At time $n \geq 2$

- Sample $X_n^i \sim q(x_n|y_n, \overline{X}_{n-1}^i)$ and set $X_{1:n}^i \leftarrow \left(\overline{X}_{1:n-1}^i, X_n^i\right)$.
- Compute the weights $\alpha_n\left(X_{n-1:n}^i\right) = \frac{g\left(y_n|X_n^i\right) f\left(X_n^i|X_{n-1}^i\right) \widetilde{p}\left(y_{n+1}|X_n^i\right)}{\widetilde{p}\left(y_n|X_{n-1}^i\right) q\left(X_n^i|y_n, X_{n-1}^i\right)}$ and $W_n^i \propto \alpha_n\left(X_{n-1:n}^i\right)$.
- Resample $\left\{W_n^i, X_{1:n}^i\right\}$ to obtain N new equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_{1:n}^i\right\}$.

Keeping in mind that this algorithm does not approximate the distributions $\{p(x_{1:n}|y_{1:n})\}$ but the distributions $\{\widetilde{p}(x_{1:n}|y_{1:n+1})\}$, we use IS to obtain an approximation of $p(x_{1:n}|y_{1:n})$ with

$$\pi_{n-1}(x_{1:n-1})\, q_n(x_n|x_{1:n-1}) = \widetilde{p}(x_{1:n-1}|y_{1:n})\, q(x_n|y_n, x_{n-1})$$

as the importance distribution. A Monte Carlo approximation of this importance distribution is obtained after the sampling step in the APF and the associated unnormalised importance weights are given by

$$\widetilde{w}_n(x_{n-1:n}) = \frac{p(x_{1:n}, y_{1:n})}{\gamma_{n-1}(x_{1:n-1})\, q_n(x_n|x_{1:n-1})} = \frac{g(y_n|x_n)\, f(x_n|x_{n-1})}{\widetilde{p}(y_n|x_{n-1})\, q(x_n|y_n, x_{n-1})}. \tag{43}$$

It follows that we obtain

$$\begin{aligned}\widehat{p}(x_{1:n}|y_{1:n}) &= \sum_{i=1}^N \widetilde{W}_n^i \delta_{X_{1:n}^i}(x_{1:n}), \\ \widehat{p}(y_{1:n}) &= \frac{1}{N}\sum_{i=1}^N \widetilde{w}_n\left(X_{n-1:n}^i\right)\end{aligned}$$

where

$$\widetilde{W}_n^i \propto \widetilde{w}_n\left(X_{n-1:n}^i\right)$$

or $\widetilde{W}_n^i \propto W_{n-1}^i \widetilde{w}_n\left(X_{n-1:n}^i\right)$ if resampling was not performed at the end of the previous iteration. Selecting $q_n(x_n|x_{1:n-1}) = p(x_n|y_n, x_{n-1})$ and $\widetilde{p}(y_n|x_{n-1}) = p(y_n|x_{n-1})$, when it is possible to do so,

leads to the so-called "perfect adaptation" case [33]. In this case, the APF takes a particularly simple form as $\alpha_n(x_{n-1:n}) = p(y_n|x_{n-1})$ and $\widetilde{w}_n(x_{n-1:n}) = 1$. This is similar to the algorithm discussed in the previous subsection where the order of the sampling and resampling steps is interchanged.

This simple reinterpretation of the APF shows us several things:

- We should select a distribution $\widetilde{p}(x_{1:n}|y_{1:n})$ with thicker tails than $p(x_{1:n}|y_{1:n})$.
- Setting $\widetilde{p}(y_n|x_{n-1}) = g(y_n|\mu(x_{n-1}))$ where μ denotes some point estimate is perhaps unwise as this will not generally satisfy that requirement.
- We should use an approximation of the predictive likelihood which is compatible with the model we are using in the sense that it encodes at least the same degree of uncertainty as the exact model.

These observations follows from the fact that $\widetilde{p}(x_{1:n}|y_{1:n})$ is used as an importance distribution to estimate $p(x_{1:n}|y_{1:n})$ and this is the usual method to ensure that the estimator variance remains finite. Thus $\widetilde{p}(y_n|x_{n-1})$ should be more diffuse than $p(y_n|x_{n-1})$.

It has been suggested in the literature to set $\widetilde{p}(y_n|x_{n-1}) = g(y_n|\mu(x_{n-1}))$ where $\mu(x_{n-1})$ corresponds to the mode, mean or median of $f(x_n|x_{n-1})$. However, this simple approximation will often yield an importance weight function (43) which is not upper bounded on $\mathcal{X} \times \mathcal{X}$ and could lead to estimates with a large/infinite variance.

An alternative approach, selecting an approximation $\widetilde{p}(y_n, x_n|x_{n-1}) = \widetilde{p}(y_n|x_{n-1})\, q(x_n|y_n, x_{n-1})$ of the distribution $p(y_n, x_n|x_{n-1}) = p(y_n|x_{n-1})\, p(x_n|y_n, x_{n-1}) = g(y_n|x_n)\, f(x_n|x_{n-1})$ such that the ratio (43) is upper bounded on $\mathcal{X} \times \mathcal{X}$ and such that it is possible to compute $\widetilde{p}(y_n|x_{n-1})$ pointwise and to sample from $q(x_n|y_n, x_{n-1})$, should be preferred.

4.3 Limitations of Particle Filters

The algorithms described earlier suffer from several limitations. It is important to emphasise at this point that, even if the optimal importance distribution $p(x_n|y_n, x_{n-1})$ can be used, this does not guarantee that the SMC algorithms will be efficient. Indeed, if the variance of $p(y_n|x_{n-1})$ is high, then the variance of the resulting approximation will be high. Consequently, it will be necessary to resample very frequently and the particle approximation $\widehat{p}(x_{1:n}|y_{1:n})$ of the joint distribution $p(x_{1:n}|y_{1:n})$ will be unreliable. In particular, for $k \ll n$ the marginal distribution $\widehat{p}(x_{1:k}|y_{1:n})$ will only be approximated by a few if not a single unique particle because the algorithm will have resampled many times between times k and n. One major problem with the approaches discussed above is that only the variables $\{X_n^i\}$ are sampled at time n but the path values $\{X_{1:n-1}^i\}$ remain fixed. An obvious way to improve upon these algorithms would involve not only sampling $\{X_n^i\}$ at time n, but also modifying the values of the paths over a fixed lag $\{X_{n-L+1:n-1}^i\}$ for $L > 1$ in light of the new observation y_n; L being fixed or upper bounded to ensure that we have a sequential algorithm (i.e. one whose computational cost and storage requirements are uniformly bounded over time). The following two sections describe two approaches to limit this degeneracy problem.

4.4 Resample-Move

This degeneracy problem has historically been addressed most often using the Resample-Move algorithm [20]. Like Markov Chain Monte Carlo (MCMC), it relies upon Markov kernels with appropriate invariant distributions. Whilst MCMC uses such kernels to generate collections of correlated samples, the Resample-Move algorithm uses them within an SMC algorithm as a principled way to "jitter" the particle locations and thus to reduce degeneracy. A Markov kernel $K_n(x'_{1:n}|x_{1:n})$ of invariant distribution

$p(x_{1:n}|y_{1:n})$ is a Markov transition kernel with the property that

$$\int p(x_{1:n}|y_{1:n}) K_n(x'_{1:n}|x_{1:n}) dx_{1:n} = p(x'_{1:n}|y_{1:n}).$$

For such a kernel, if $X_{1:n} \sim p(x_{1:n}|y_{1:n})$ and $X'_{1:n}|X_{1:n} \sim K(x_{1:n}|X_{1:n})$ then $X'_{1:n}$ is still marginally distributed according to $p(x_{1:n}|y_{1:n})$. Even if $X_{1:n}$ is not distributed according to $p(x_{1:n}|y_{1:n})$ then, after an application of the MCMC kernel, $X'_{1:n}$ can only have a distribution closer than that of $X_{1:n}$ (in total variation norm) to $p(x_{1:n}|y_{1:n})$. A Markov kernel is said to be ergodic if iterative application of that kernel generates samples whose distribution converges towards $p(x_{1:n}|y_{1:n})$ irrespective of the distribution of the initial state.

It is easy to construct a Markov kernel with a specified invariant distribution. Indeed, this is the basis of MCMC — for details see [35] and references therein. For example, we could consider the following kernel, based upon the Gibbs sampler: set $x'_{1:n-L} = x_{1:n-L}$ then sample x'_{n-L+1} from $p\left(x_{n-L+1}|y_{1:n}, x'_{1:n-L}, x_{n-L+2:n}\right)$, sample x'_{n-L+2} from $p\left(x_{n-L+2}|y_{1:n}, x'_{1:n-L+1}, x_{n-L+3:n}\right)$ and so on until we sample x'_n from $p\left(x_n|y_{1:n}, x'_{1:n-1}\right)$; that is

$$K_n(x'_{1:n}|x_{1:n}) = \delta_{x_{1:n-L}}\left(x'_{1:n-L}\right) \prod_{k=n-L+1}^{n} p\left(x'_k|y_{1:n}, x'_{1:k-1}, x_{k+1:n}\right)$$

and we write, with a slight abuse of notation, the non-degenerate component of the MCMC kernel $K_n\left(x'_{n-L+1:n}|x_{1:n}\right)$. It is straightforward to verify that this kernel is $p(x_{1:n}|y_{1:n})$-invariant.

If it is not possible to sample from $p\left(x'_k|y_{1:n}, x'_{1:k-1}, x_{k+1:n}\right) = p\left(x'_k|y_k, x'_{k-1}, x_{k+1}\right)$, we can instead employ a Metropolis-Hastings (MH) strategy and sample a candidate according to some proposal $q\left(x'_k|y_k, x'_{k-1}, x_{k:k+1}\right)$ and accept it with the usual MH acceptance probability

$$\begin{aligned}
&\min\left(1, \frac{p(x'_{1:k}, x_{k+1:n}|y_{1:n})\, q\left(x_k|y_k, x'_{k-1}, x'_k, x_{k+1}\right)}{p\left(x'_{1:k-1}, x_{k+1:n}|y_{1:n}\right) q\left(x'_k|y_k, x'_{k-1}, x_{k:k+1}\right)}\right) \\
&= \min\left(1, \frac{g(y_k|x'_k) f(x_{k+1}|x'_k) f\left(x'_k|x'_{k-1}\right) q\left(x_k|y_k, x'_{k-1}, x'_k, x_{k+1}\right)}{g(y_k|x_k) f(x_{k+1}|x_k) f\left(x_k|x'_{k-1}\right) q\left(x'_k|y_k, x'_{k-1}, x_{k:k+1}\right)}\right).
\end{aligned}$$

It is clear that these kernels can be ergodic only if $L = n$ and *all* of the components of $x_{1:n}$ are updated. However, in our context we will typically not use ergodic kernels as this would require sampling an increasing number of variables at each time step. In order to obtain truly online algorithms, we restrict ourselves to updating the variables $X_{n-L+1:n}$ for some fixed or bounded L.

The algorithm proceeds as follows, with K_n denoting a Markov kernel of invariant distribution $p(x_{1:n}|y_{1:n})$.

SMC Filtering with MCMC Moves

At time $n = 1$

- Sample $X_1^i \sim q(x_1|y_1)$.
- Compute the weights $w_1\left(X_1^i\right) = \frac{\mu\left(X_1^i\right) g\left(y_1|X_1^i\right)}{q\left(X_1^i|y_1\right)}$ and $W_1^i \propto w_1\left(X_1^i\right)$.
- Resample $\left\{W_1^i, X_1^i\right\}$ to obtain N equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_1^i\right\}$.
- Sample $X_1'^i \sim K_1(x_1|\overline{X}_1^i)$.

At time $1 < n < L$

- Sample $X_n^i \sim q(x_n|\, y_n, X_{n-1}'^i)$ and set $X_{1:n}^i \leftarrow \left(X_{1:n-1}'^i, X_n^i\right)$.
- Compute the weights $\alpha_n\left(X_{n-1:n}^i\right) = \frac{g\left(y_n|X_n^i\right) f\left(X_n^i|X_{n-1}^i\right)}{q\left(X_n^i|y_n, X_{n-1}^i\right)}$ and $W_n^i \propto \alpha_n\left(X_{n-1:n}^i\right)$.
- Resample $\left\{W_n^i, X_{1:n}^i\right\}$ to obtain N equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_{1:n}^i\right\}$.
- Sample $X_{1:n}'^i \sim K_n(x_{1:n}|\, \overline{X}_{1:n}^i)$.

At time $n \geq L$

- Sample $X_n^i \sim q(x_n|\, y_n, X_{n-1}'^i)$ and set $X_{1:n}^i \leftarrow \left(X_{1:n-1}'^i, X_n^i\right)$.
- Compute the weights $\alpha_n\left(X_{n-1:n}^i\right) = \frac{g\left(y_n|X_n^i\right) f\left(X_n^i|X_{n-1}^i\right)}{q\left(X_n^i|y_n, X_{n-1}^i\right)}$ and $W_n^i \propto \alpha_n\left(X_{n-1:n}^i\right)$.
- Resample $\left\{W_n^i, X_{1:n}^i\right\}$ to obtain N new equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_{1:n}^i\right\}$.
- Sample $X_{n-L+1:n}'^i \sim K_n(x_{n-L+1:n}|\, \overline{X}_{1:n}^i)$ and set $X_{1:n}'^i \leftarrow \left(\overline{X}_{1:n-L}^i, X_{n-L+1:n}'^i\right)$.

The following premise, which [35] describes as "generalised importance sampling", could be used to justify inserting MCMC transitions into an SMC algorithm after the sampling step. Given a target distribution π, an instrumental distribution μ and a π-invariant Markov kernel, K, the following generalisation of the IS identity holds:

$$\int \pi(y)\varphi(y)dy \quad = \quad \iint \mu(x)K(\,y|\,x)\frac{\pi(y)L(\,x|\,y)}{\mu(x)K(\,y|\,x)}\varphi(y)dxdy$$

for any Markov kernel L. This approach corresponds to importance sampling on an enlarged space using $\mu(x)K(\,y|\,x)$ as the proposal distribution for a target $\pi(y)L(\,x|\,y)$ and then estimating a function $\varphi'(x,y) = \varphi(y)$. In particular, for the time-reversal kernel associated with K

$$L(\,x|\,y) = \frac{\pi(x)K(\,y|\,x)}{\pi\,(y)},$$

we have the importance weight

$$\frac{\pi(y)L(\,x|\,y)}{\mu(x)K(\,y|\,x)} = \frac{\pi\,(x)}{\mu(x)}.$$

This interpretation of such an approach illustrates its deficiency: the importance weights depend only upon the location before the MCMC move while the sample depends upon the location after the move. Even if the kernel was perfectly mixing, leading to a collection of iid samples from the target distribution, some of these samples would be eliminated and some replicated in the resampling step. Resampling before an MCMC step will always lead to greater sample diversity than performing the steps in the other order (and this algorithm can be justified directly by the invariance property).

Based on this reinterpretation of MCMC moves within IS, it is possible to reformulate this algorithm as a specific application of the generic SMC algorithm discussed in Section 3. To simply notation we write $q_n\left(x_n|\, x_{n-1}\right)$ for $q\left(x_n|\, y_n, x_{n-1}\right)$. To clarify our argument, it is necessary to add a superscript to the variables; e.g. X_k^p corresponds to the p^{th} time the random variable X_k is sampled; in this and the following section, this superscript *does not* denote the particle index. Using such notation, this algorithm

is the generic SMC algorithm associated to the following sequence of target distributions

$$\begin{aligned}
&\pi_n\left(x_1^{1:L+1},\ldots,x_{n-L+1}^{1:L+1},x_{n-L}^{1:L},...,x_n^{1:2}\right)\\
&=p\left(x_1^{L+1},\ldots.,x_{n-L+1}^{L+1},x_{n-L}^{L},...,x_n^{2}\middle|\, y_{1:n}\right)L_n\left(x_n^1,x_{n-1}^2,...,x_{n-L+1}^L\middle|\, x_n^2,x_{n-1}^3,...,x_{n-L+1}^{L+1}\right)\\
&\times\cdots\times L_2\left(x_1^2,x_2^1\middle|\, x_1^3,x_2^2\right)L_1\left(x_1^1\middle|\, x_1^2\right)
\end{aligned}$$

where L_n is the time-reversal kernel associated with K_n whereas, if no resampling is used[5], a path up to time n is sampled according to

$$\begin{aligned}
&q_n\left(x_1^{1:L+1},\ldots,x_{n-L+1}^{1:L+1},x_{n-L}^{1:L},...,x_n^{1:2}\right)\\
&=q_1\left(x_1^1\right)K_1\left(x_1^2\middle|\, x_1^1\right)q_2\left(x_2^1\middle|\, x_1^2\right)K_2\left(x_1^3,x_2^2\middle|\, x_1^2,x_2^1\right)\\
&\times\cdots\times q_n\left(x_n^1\middle|\, x_{n-1}^2\right)K_n\left(x_{n-L+1}^{L+1},\ldots,x_{n-1}^3,x_n^2\middle|\, x_{1:n-L}^{L+1},x_{n-L+1}^L,\ldots,x_{n-1}^2,x_n^1\right).
\end{aligned}$$

This sequence of target distributions admits the filtering distributions of interest as marginals. The clear theoretical advantage of using MCMC moves is that the use of even non-ergodic MCMC kernels $\{K_n\}$ can only improve the mixing properties of $\{\pi_n\}$ compared to the "natural" sequence of filtering distributions; this explains why these algorithms outperform a standard particle filter for a given number of particles.

Finally, we note that the incorporation of MCMC moves to improve sample diversity is an idea which is appealing in its simplicity and which can easily be incorporated into any of the algorithms described here.

4.5 Block Sampling

The Resample-Move method discussed in the previously section suffers from a major drawback. Although it does allow us to reintroduce some diversity among the set of particles after the resampling step over a lag of length L, the importance weights have the same expression as for the standard particle filter. So this strategy does not significantly decrease the number of resampling steps compared to a standard approach. It can partially mitigate the problem associated with resampling, but it does not prevent these resampling steps in the first place.

An alternative *block sampling* approach has recently been proposed in [15]. This approach goes further than the Resample-Move method, which aims to sample only the component x_n at time n in regions of high probability mass and then to uses MCMC moves to rejuvenate $x_{n-L+1:n}$ after a resampling step. The block sampling algorithm attempts to directly sample the components $x_{n-L+1:n}$ at time n; the previously-sampled values of the components $x_{n-L+1:n-1}$ sampled are simply discarded. In this case, it can easily be shown that the optimal importance distribution (that which minimises the variance of the importance weights at time n) is:

$$p\left(x_{n-L+1:n}\middle|\, y_{n-L+1:n},x_{n-L}\right)=\frac{p\left(x_{n-L:n},y_{n-L+1:n}\right)}{p\left(y_{n-L+1:n}\middle|\, x_{n-L}\right)} \tag{44}$$

where

$$p\left(y_{n-L+1:n}\middle|\, x_{n-L}\right)=\int\prod_{k=n-L+1}^{n}f\left(x_k\middle|\, x_{k-1}\right)\cdot g\left(y_k\middle|\, x_k\right)dx_{n-L+1:n}. \tag{45}$$

As in the standard case (corresponding to $L=1$), it is typically impossible to sample from (44) and/or to compute (45). So in practice we need to design an importance distribution $q\left(x_{n-L+1:n}\middle|\, y_{n-L+1:n},x_{n-L}\right)$ approximating $p\left(x_{n-L+1:n}\middle|\, y_{n-L+1:n},x_{n-L}\right)$. Henceforth, we consider the case where $L>1$.

The algorithm proceeds as follows.

[5] Once again, similar expressions can be obtained in the presence of resampling and the technique remains valid.

SMC Block Sampling for Filtering

At time $n = 1$

- Sample $X_1^i \sim q(x_1 | y_1)$.
- Compute the weights $w_1\left(X_1^i\right) = \frac{\mu\left(X_1^i\right) g\left(y_1 | X_1^i\right)}{q\left(X_1^i | y_1\right)}$ and $W_1^i \propto w_1\left(X_1^i\right)$.
- Resample $\left\{W_1^i, X_1^i\right\}$ to obtain N equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_1^i\right\}$.

At time $1 < n < L$

- Sample $X_{1:n}^i \sim q(x_{1:n} | y_{1:n})$.
- Compute the weights $\alpha_n\left(X_{1:n}^i\right) = \frac{p\left(X_{1:n}^i, y_{1:n}\right)}{q\left(X_{1:n}^i | y_{1:n}\right)}$ and $W_n^i \propto \alpha_n\left(X_{n-1:n}^i\right)$.
- Resample $\left\{W_n^i, X_{1:n}^i\right\}$ to obtain N equally-weighted particles $\left\{\frac{1}{N}, \overline{X}_{1:n}^i\right\}$.

At time $n \geq L$

- Sample $X_{n-L+1:n}^i \sim q(x_{n-L+1:n} | y_{n-L+1:n}, \overline{X}_{1:n-1}^i)$.
- Compute the weights

$$w_n\left(\overline{X}_{n-L:n-1}^i, X_{n-L+1:n}^i\right) = \frac{p\left(\overline{X}_{1:n-L}^i, X_{n-L+1:n}^i, y_{1:n}\right) q(\overline{X}_{n-L+1:n-1}^i | y_{n-L+1:n-1}, \overline{X}_{n-L}^i)}{p\left(\overline{X}_{1:n-1}^i, y_{1:n-1}\right) q(X_{n-L+1:n}^i | y_{n-L+1:n}, \overline{X}_{n-L}^i)} \tag{46}$$

and $W_n^i \propto w_n\left(\overline{X}_{n-L:n-1}^i, X_{n-L+1:n}^i\right)$.

- Resample $\left\{W_n^i, \overline{X}_{1:n-L}^i, X_{n-L+1:n}^i\right\}$ to obtain N new equally weighted particles $\left\{\frac{1}{N}, \overline{X}_{1:n}^i\right\}$.

When the optimal IS distribution is used $q\left(x_{n-L+1:n} | y_{n-L+1:n}, x_{n-L}\right) = p\left(x_{n-L+1:n} | y_{n-L+1:n}, x_{n-L}\right)$, we obtain

$$\begin{aligned} w_n\left(\overline{x}_{1:n-1}, x_{n-L+1:n}\right) &= \frac{p\left(\overline{x}_{1:n-L}, x_{n-L+1:n}, y_{1:n}\right) p(\overline{x}_{n-L+1:n-1} | y_{n-L+1:n-1}, \overline{x}_{n-L})}{p\left(\overline{x}_{1:n-1}, y_{1:n-1}\right) p\left(x_{n-L+1:n} | y_{n-L+1:n}, \overline{x}_{n-L}\right)} \\ &= p\left(y_n | y_{n-L+1:n}, \overline{x}_{n-L}\right). \end{aligned}$$

This optimal weight has a variance which typically decreases exponentially fast with L (under mixing assumptions). Hence, in the context of adaptive resampling, this strategy dramatically reduces the number of resampling steps. In practice, we cannot generally compute this optimal weight and thus use (46) with an approximation of $p\left(x_{n-L+1:n} | y_{n-L+1:n}, x_{n-L}\right)$ for $q\left(x_{n-L+1:n} | y_{n-L+1:n}, x_{n-L}\right)$. When a good approximation is available, the variance of (46) can be reduced significantly compared to the standard case where $L = 1$.

Again, this algorithm is a specific application of the generic SMC algorithm discussed in Section 3. To simply notation we write $q_n\left(x_{n-L+1:n} | x_{n-L}\right)$ for $q\left(x_{n-L+1:n} | y_{n-L+1:n}, x_{n-L}\right)$. To clarify our argument, it is again necessary to add a superscript to the variables; e.g. X_k^p corresponds to the p^{th} time the random variable X_k is sampled; remember that in the present section, this superscript *does not* denote the particle index. Using this notation, the algorithm corresponds to the generic SMC algorithm associated with the

following sequence of target distributions

$$\begin{aligned}&\pi_n\left(x_1^{1:L},\ldots,x_{n-L+1}^{1:L},x_{n-L+2}^{1:L-1},\ldots,x_n^1\right)\\&=p\left(x_{1:n-L+1}^L,x_{n-L+2}^{L-1},\ldots,x_n^1\middle|\, y_{1:n}\right)q_{n-1}\left(x_{n-L+1}^{L-1},\ldots,x_{n-1}^1\middle|\, x_{n-L}^L\right)\\&\times\cdots q_2\left(x_1^2,x_2^1\right)q_1\left(x_1^1\right).\end{aligned}$$

where, if no resampling is used, a path is sampled according to

$$\begin{aligned}&q_n\left(x_1^{1:L},\ldots,x_{n-L+1}^{1:L},x_{n-L+2}^{1:L-1},...,x_n^1\right)\\&=q_1\left(x_1^1\right)q_2\left(x_1^2,x_2^1\right)\times\cdots\times q_n\left(x_{n-L+1}^L,...,x_n^1\middle|\, x_{n-L}^L\right).\end{aligned}$$

The sequence of distributions admits the filtering distributions of interest as marginals. The mixing properties of $\{\pi_n\}$ are also improved compared to the 'natural' sequence of filtering distributions; this is the theoretical explanation of the better performance obtained by these algorithms for a given number of particles.

4.6 Rao-Blackwellised Particle Filtering

Let us start by quoting Trotter [36]: "A good Monte Carlo is a dead Monte Carlo". Trotter specialised in Monte Carlo methods and did not advocate that we should not use them, but that we should avoid them whenever possible. In particular, whenever an integral can be calculated analytically doing so should be preferred to the use of Monte Carlo techniques.

Assume, for example, that one is interested in sampling from $\pi(x)$ with $x=(u,z)\in\mathcal{U}\times\mathcal{Z}$ and

$$\pi(x)=\frac{\gamma(u,z)}{Z}=\pi(u)\,\pi(z|\,u)$$

where $\pi(u)=Z^{-1}\gamma(u)$ and

$$\pi(z|\,u)=\frac{\gamma(u,z)}{\gamma(u)}$$

admits a closed-form expression; e.g. a multivariate Gaussian. Then if we are interested in approximating $\pi(x)$ and computing Z, we only need to perform a MC approximation $\pi(u)$ and $Z=\int\gamma(u)\,du$ on the space $\mathcal{U}$ instead of $\mathcal{U}\times\mathcal{Z}$. We give two classes of important models where this simple idea can be used successfully.

4.6.1 Conditionally linear Gaussian models

Consider $\mathcal{X}=\mathcal{U}\times\mathcal{Z}$ with $\mathcal{Z}=\mathbb{R}^{n_z}$. Here $X_n=(U_n,Z_n)$ where $\{U_n\}$ is a unobserved Markov process such that $U_1\sim\mu_U(u_1)$, $U_n|\,U_{n-1}\sim f_U(u_n|\,U_{n-1})$ and conditional upon $\{U_n\}$ we have a linear Gaussian model with $Z_1|\,U_1\sim\mathcal{N}(0,\Sigma_{U_1})$ and

$$\begin{aligned}Z_n&=A_{U_n}Z_{n-1}+B_{U_n}V_n,\\Y_n&=C_{U_n}Z_n+D_{U_n}W_n\end{aligned}$$

where $V_n\overset{\text{i.i.d.}}{\sim}\mathcal{N}(0,I_{n_v})$, $W_n\overset{\text{i.i.d.}}{\sim}\mathcal{N}(0,I_{n_w})$ and for any $u\in\mathcal{U}$ $\{A_u,B_u,C_u,D_u\}$ are matrices of appropriate dimensions. In this case we have $\mu(x)=\mu(u,z)=\mu_U(u)\mathcal{N}(x;0,\Sigma_z)$, $f(x'|\,x)=f((u',z')|\,(u,z))=f_U(u'|\,u)\mathcal{N}\left(z';A_{u'}z,B_{u'}B_{u'}^{\mathrm{T}}\right)$ and $g(y|\,x)=g(y|\,(u,z))=\mathcal{N}\left(y;C_uz,D_uD_u^{\mathrm{T}}\right)$. The switching state-space model discussed in *Example* 3 corresponds to the case where $\{U_n\}$ is a finite state-space Markov process.

We are interested in estimating

$$p(u_{1:n},z_{1:n}|\,y_{1:n})=p(u_{1:n}|\,y_{1:n})\,p(z_{1:n}|\,y_{1:n},u_{1:n}).$$

Conditional upon $\{U_n\}$, we have a standard linear Gaussian model $\{Z_n\}$, $\{Y_n\}$. Hence $p(z_{1:n}|y_{1:n}, u_{1:n})$ is a Gaussian distribution whose statistics can be computed using Kalman techniques; e.g. the marginal $p(z_n|y_{1:n}, u_{1:n})$ is a Gaussian distribution whose mean and covariance can be computed using the Kalman filter. It follows that we only need to use particle methods to approximate

$$\begin{aligned}\gamma_n(u_{1:n}) &= p(u_{1:n}, y_{1:n}) \\ &= p(u_{1:n})\, p(y_{1:n}|u_{1:n})\end{aligned}$$

where $p(u_{1:n})$ follows from the Markov assumption on $\{U_n\}$ and $p(y_{1:n}|u_{1:n})$ is a marginal likelihood which can be computed through the Kalman filter. In this case, we have

$$\begin{aligned}q^{\text{opt}}(u_n|y_{1:n}, u_{1:n-1}) &= p(u_n|y_{1:n}, u_{1:n-1}) \\ &= \frac{p(y_n|y_{1:n-1}, u_{1:n})\, f_U(u_n|u_{n-1})}{p(y_n|y_{1:n-1}, u_{1:n-1})}.\end{aligned}$$

The standard SMC algorithm associated with $\gamma_n(u_{1:n})$ and the sequence of IS distributions $q(u_n|y_{1:n}, u_{1:n-1})$ proceeds as follows.

SMC for Filtering in Conditionally Linear Gaussian Models

At time $n = 1$

- Sample $U_1^i \sim q(u_1|y_1)$.
- Compute the weights $w_1(U_1^i) = \frac{\mu_U(U_1^i)p(y_1|U_1^i)}{q(U_1^i|y_1)}$ and $W_1^i \propto w_1(U_1^i)$.
- Resample $\{W_1^i, U_1^i\}$ to obtain N equally-weighted particles $\left\{\frac{1}{N}, \overline{U}_1^i\right\}$.

At time $n \geq 2$

- Sample $U_n^i \sim q\left(u_n|y_{1:n}, \overline{U}_{1:n-1}^i\right)$ and set $U_{1:n}^i \leftarrow \left(\overline{U}_{1:n-1}^i, U_n^i\right)$.
- Compute the weights $\alpha_n(U_{1:n}^i) = \frac{p(y_n|y_{1:n-1}, U_{1:n}^i) f_U(U_n^i|U_{n-1}^i)}{q(U_n^i|y_{1:n}, U_{1:n-1}^i)}$ and $W_n^i \propto \alpha_n(U_{1:n}^i)$.
- Resample $\{W_n^i, U_{1:n}^i\}$ to obtain N new equally-weighted particles $\left\{\frac{1}{N}, \overline{U}_{1:n}^i\right\}$.

This algorithm provides also two approximations of $p(u_{1:n}|y_{1:n})$ given by

$$\widehat{p}(u_{1:n}|y_{1:n}) = \sum_{i=1}^N W_n^i \delta_{U_{1:n}^i}(u_{1:n}),$$

$$\overline{p}(u_{1:n}|y_{1:n}) = \frac{1}{N}\sum_{i=1}^N \delta_{\overline{U}_{1:n}^i}(u_{1:n})$$

and

$$\widehat{p}(y_n|y_{1:n-1}) = \frac{1}{N}\sum_{i=1}^N \alpha_n(U_{1:n}^i).$$

At first glance, it seems that this algorithm cannot be implemented as it requires storing paths $\{U_{1:n}^i\}$ of increasing dimension so as to allow the computation of $p(y_n|y_{1:n-1}, u_{1:n})$ and sampling from $q(u_n|y_{1:n}, u_{1:n-1})$.

The key is to realise that $p(y_n|y_{1:n-1}, u_{1:n})$ is a Gaussian distribution of mean $y_{n|n-1}(u_{1:n})$ and covariance $S_{n|n-1}(u_{1:n})$ which can be computed using the Kalman filter. Similarly, given that the optimal IS distribution only depends on the path $u_{1:n}$ through $p(y_n|y_{1:n-1}, u_{1:n})$, it is sensible to build an importance distribution $q(u_n|y_{1:n}, u_{1:n-1})$ which only depends on $u_{1:n}$ through $p(y_n|y_{1:n-1}, u_{1:n})$. Hence in practice, we do not need to store $\{U^i_{1:n}\}$ but only $\{U^i_{n-1:n}\}$ and the Kalman filter statistics associated with $\{U^i_{1:n}\}$. The resulting particle filter is a bank of interacting Kalman filters where each Kalman filter is used to compute the marginal likelihood term $p(y_{1:n}|u_{1:n})$.

4.6.2 Partially observed linear Gaussian models

The same idea can be applied to the class of partially observed linear Gaussian models [2]. Consider $\mathcal{X} = \mathcal{U} \times \mathcal{Z}$ with $\mathcal{Z} = \mathbb{R}^{n_z}$. Here $X_n = (U_n, Z_n)$ with $Z_1 \sim \mathcal{N}(0, \Sigma_1)$

$$\begin{aligned} Z_n &= AZ_{n-1} + BV_n, \\ U_n &= CZ_n + DW_n \end{aligned}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{n_v})$, $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{n_w})$; see [2] for generalisations. We make the additional assumption that

$$g(y_n|x_n) = g(y_n|(u_n, z_n)) = g(y_n|u_n).$$

In this case, we are interested in estimating

$$\begin{aligned} p(u_{1:n}, z_{1:n}|y_{1:n}) &= p(u_{1:n}|y_{1:n})\, p(z_{1:n}|y_{1:n}, u_{1:n}) \\ &= p(u_{1:n}|y_{1:n})\, p(z_{1:n}|u_{1:n}) \end{aligned}$$

where $p(z_{1:n}|u_{1:n})$ is a multivariate Gaussian distribution whose statistics can be computed using a Kalman filter associated with the linear model $\{U_n, Z_n\}$. It follows that we only need to use particle methods to approximate

$$\begin{aligned} \gamma_n(u_{1:n}) &= p(u_{1:n}, y_{1:n}) \\ &= p(u_{1:n})\, p(y_{1:n}|u_{1:n}) \end{aligned}$$

where $p(y_{1:n}|u_{1:n}) = \prod_{k=1}^{n} g(y_k|u_k)$ and $p(u_{1:n})$ is the marginal Gaussian prior of $\{U_n\}$ which corresponds to the 'marginal' likelihood term which can be computed using the Kalman filter associated with $\{U_n, Z_n\}$. The resulting particle filter is also an interacting bank of Kalman filters but the Kalman filters are here used to compute the marginal prior $p(u_{1:n})$.

5 Particle Smoothing

We have seen previously that SMC methods can provide an approximation of the sequence of distributions $\{p(x_{1:n}|y_{1:n})\}$. Consequently, sampling from a joint distribution $p(x_{1:T}|y_{1:T})$ and approximating the marginals $\{p(x_n|y_{1:T})\}$ for $n = 1, ..., T$ is straightforward. We just run an SMC algorithm up to time T and sample from/marginalise our SMC approximation $\widehat{p}(x_{1:T}|y_{1:T})$. However, we have seen that this approach is bound to be inefficient when T is large as the successive resampling steps lead to particle degeneracy: $\widehat{p}(x_{1:n}|y_{1:T})$ is approximated by a single unique particle for $n \ll T$. In this section, we discuss various alternative schemes which do not suffer from these problems. The first method relies on an simple fixed-lag approximation whereas the other algorithms rely on the forward-backward recursions presented in Section 2.3.

5.1 Fixed-lag Approximation

The fixed-lag approximation is the simplest approach. It was proposed in [26]. It relies on the fact that, for hidden Markov models with "good" forgetting properties, we have

$$p\left(x_{1:n}|\, y_{1:T}\right) \approx p\left(x_{1:n}|\, y_{1:\min(n+\Delta,T)}\right) \tag{47}$$

for Δ large enough; that is observations collected at times $k > n + \Delta$ do not bring any additional information about $x_{1:n}$. This suggests a very simple scheme — simply don't update the estimate at time k after time $k = n + \Delta$. Indeed, in practice we just do not resample the components $X^i_{1:n}$ of the particles $X^i_{1:k}$ at times $k > n + \Delta$. This algorithm is trivial to implement but the main practical problem is that we typically do not know Δ. Hence we need to replace Δ with an estimate of it denoted L. If we select $L < \Delta$, then $p\left(x_{1:n}|\, y_{1:\min(n+L,T)}\right)$ is a poor approximation of $p\left(x_{1:n}|\, y_{1:T}\right)$. If we select a large values of L to ensure that $L \geq \Delta$ then the degeneracy problem remains substantial. Unfortunately automatic selection of L is difficult (and, of course, for some poorly-mixing models Δ is so large that this approach is impractical). Experiments on various models have shown that good performance were achieved with $L \approx 20 - 50$. Note that such fixed-lag SMC schemes do not converge asymptotically (i.e. as $N \to \infty$) towards the true smoothing distributions because we do not have $p\left(x_{1:n}|\, y_{1:T}\right) = p\left(x_{1:n}|\, y_{1:\min(n+L,T)}\right)$. However, the bias might be negligible and can be upper bounded under mixing conditions [8]. It should also be noted that this method does not provide an approximation of the joint $p\left(x_{1:T}|\, y_{1:T}\right)$ — it approximates only the marginal distributions.

5.2 Forward Filtering-Backward Smoothing

We have seen previously that it is possible to sample from $p\left(x_{1:T}|\, y_{1:T}\right)$ and compute the marginals $\{p\left(x_n|\, y_{1:T}\right)\}$ using forward-backward formulæ. It is possible to obtain an SMC approximation of the forward filtering-backward sampling procedure directly by noting that for

$$\widehat{p}\left(x_n|\, y_{1:n}\right) = \sum_{i=1}^{N} W^i_n \delta_{X^i_n}\left(x_n\right)$$

we have

$$\begin{aligned}\widehat{p}\left(x_n|\, X_{n+1}, y_{1:n}\right) &= \frac{f\left(X_{n+1}|\, x_n\right)\widehat{p}\left(x_n|\, y_{1:n}\right)}{\int f\left(X_{n+1}|\, x_n\right)\widehat{p}\left(x_n|\, y_{1:n}\right)dx_n} \\ &= \sum_{i=1}^{N} \frac{W^i_n f\left(X_{n+1}|X^i_n\right)\delta_{X^i_n}\left(x_n\right)}{\sum_{j=1}^{N} W^j_n f\left(X_{n+1}|X^j_n\right)}.\end{aligned}$$

Consequently, the following algorithm generates a sample approximately distributed according to $p\left(x_{1:T}|\, y_{1:T}\right)$: first sample $X_T \sim \widehat{p}\left(x_T|\, y_{1:T}\right)$ and for $n = T-1, T-2, ..., 1$, sample $X_n \sim \widehat{p}\left(x_n|\, X_{n+1}, y_{1:n}\right)$.

Similarly, we can also provide an SMC approximation of the forward filtering-backward smoothing procedure by direct means. If we denote by

$$\widehat{p}(x_n|y_{1:T}) = \sum_{i=1}^{N} W^i_{n|T}\delta_{X^i_n}(x_n) \tag{48}$$

the particle approximation of $p(x_n|y_{1:T})$ then, by inserting (48) into (15), we obtain

$$\begin{aligned}\widehat{p}(x_n|y_{1:T}) &= \sum_{i=1}^{N} W^i_n \left[\sum_{j=1}^{N} W^j_{n+1|T} \frac{f\left(X^j_{n+1}|X^i_n\right)}{\left[\sum_{l=1}^{N} W^l_n f\left(X^j_{n+1}|X^l_n\right)\right]}\right]\delta_{X^i_n}(x_n) \\ &:= \sum_{i=1}^{N} W^i_{n|T}\delta_{X^i_n}(x_n).\end{aligned} \tag{49}$$

The forward filtering-backward sampling approach requires $\mathcal{O}(NT)$ operations to sample one path approximately distributed according to $p(x_{1:T}|y_{1:T})$ whereas the forward filtering-backward smoothing algorithm requires $\mathcal{O}(N^2T)$ operations to approximate $\{p(x_n|y_{1:T})\}$. Consequently, these algorithms are only useful for very long time series in which sample degeneracy prevents computationally-cheaper, crude methods from working.

5.3 Generalised Two-filter Formula

To obtain an SMC approximation of the generalised two-filter formula (18), we need to approximate the backward filter $\{\widetilde{p}(x_n|y_{n:T})\}$ and to combine the forward filter and the backward filter in a sensible way. To obtain an approximation of $\{\widetilde{p}(x_n|y_{n:T})\}$, we simply use an SMC algorithm which targets the sequence of distributions $\{\widetilde{p}(x_{n:T}|y_{n:T})\}$ defined in (17). We run the SMC algorithm "backward in time" to approximate $\widetilde{p}(x_T|y_T)$ then $\widetilde{p}(x_{T-1:T}|y_{T-1:T})$ and so on. We obtain an SMC approximation denoted

$$\widehat{\widetilde{p}}(x_{n:T}|y_{n:T}) = \sum_{i=1}^{N} \widetilde{W}_n^i \delta_{X_{n:T}^i}(x_{n:T}).$$

To combine the forward filter and the backward filter, we obviously cannot multiply the SMC approximations of both $p(x_n|y_{1:n-1})$ and $\widetilde{p}(x_{n:T}|y_{n:T})$ directly, so we first rewrite Eq. (18) as

$$p(x_n|y_{1:T}) \propto \frac{\int f(x_n|x_{n-1})\, p(x_{n-1}|y_{1:n-1})\, dx_{n-1} \cdot \widetilde{p}(x_n|y_{n:T})}{\widetilde{p}_n(x_n)}.$$

By plugging the SMC approximations of $p(x_{n-1}|y_{1:n-1})$ and $\widetilde{p}(x_n|y_{n:T})$ in this equation, we obtain

$$\widehat{p}(x_n|y_{1:T}) = \sum_{i=1}^{N} W_{n|T}^i \delta_{\widetilde{X}_n^i}(x_n)$$

where

$$W_{n|T}^j \propto \widetilde{W}_n^j \sum_{i=1}^{N} W_{n-1}^i \frac{f\left(\widetilde{X}_n^j | X_{n-1}^i\right)}{\widetilde{p}_n\left(\widetilde{X}_n^j\right)}. \tag{50}$$

Like the SMC implementation of the forward-backward smoothing algorithm, this approach has a computational complexity $\mathcal{O}(N^2T)$. However fast computational methods have been developed to address this problem [27]. Moreover it is possible to reduce this computational complexity to $\mathcal{O}(NT)$ by using rejection sampling to sample from $p(x_n|y_{1:T})$ using $p(x_{n-1}|y_{1:n-1})$ and $\widetilde{p}(x_n|y_{n:T})$ as proposal distributions. More recently, an importance sampling type approach has also been proposed in [19] to reduce the computational complexity to $O(NT)$; see [6] for a related idea developed in the context of belief propagation. Compared to the forward-backward formula, it might be expected to substantially outperform that algorithm in any situation in which the support of the smoothed estimate differs substantially from that of the filtering estimate. That is, in those situations in which observations obtained at time $k > n$ provide a significant amount of information about the state at time n[6]. The improvement arises from the fact that the SMC implementation of the forward-backward smoother simply reweights a sample set which targets $p(x_{1:n}|y_{1:n})$ to account for the information provided by $y_{n+1:k}$ whereas the two filter approach uses a different sample set with locations appropriate to the smoothing distributions.

6 Summary

We have provided a review of particle methods for filtering, marginal likelihood computation and smoothing. Having introduced a simple SMC algorithm, we have shown how essentially all of the particle-based

[6] This situation occurs, for example, whenever observations are only weakly informative and the state evolution involves relatively little stochastic variability. An illustrative example provided in [5] shows that this technique can dramatically outperform all of the other approaches detailed above.

methods introduced in the literature to solve these problems can be interpreted as a combination of two operations: sampling and resampling. By considering an appropriate sequence of target distributions, defined on appropriate spaces, it is possible to interpret all of these algorithms as particular cases of the general algorithm.

This interpretation has two primary advantages:

1. The standard algorithm may be viewed as a particle interpretation of a Feynman-Kac model (see [11]) and hence strong, sharp theoretical results can be applied to all algorithms within this common formulation.
2. By considering all algorithms within the same framework is is possible to develop a common understanding and intuition allowing meaningful comparisons to be drawn and sensible implementation decisions to be made.

Although much progress has been made over the past fifteen years, and the algorithms described above provide good estimates in many complex, realistic settings, there remain a number of limitations and open problems:

- As with any scheme for numerical integration, be it deterministic or stochastic, there exist problems which exist on sufficiently high-dimensional spaces and involving sufficiently complex distributions that it is not possible to obtain an accurate characterisation in a reasonable amount of time.
- In many settings of interest the likelihood and state transition density are only known up to a vector of unknown parameters. It is typically of interest to estimate this vector of parameters at the same time as (or in preference to) the vector of states. Formally, such situations can be described as directly as a state space model with a degenerate transition kernel. However, this degeneracy prevents the algorithms described above from working in practice.
- Several algorithms intended specifically for parameter estimation have been developed in recent years. Unfortunately, space constraints prevent us from discussing these methods within the current tutorial. Although progress has been made, this is a difficult problem and it cannot be considered to have been solved in full generality. These issues will be discussed in further depth in an extended version of this tutorial which is currently in preparation.

It should also be mentioned that the SMC algorithm presented in this tutorial can be adapted to perform much more general Monte Carlo simulation: it is not restricted to problems of the filtering type, or even to problems with a sequential character. It has recently been established that it is also possible to employ SMC within MCMC and to obtain a variety of related algorithms.

References

[1] Anderson, B. D. O. and Moore, J. B. (1979) *Optimal Filtering*, Prentice Hall, Englewood Cliffs, New Jersey.

[2] Andrieu, C. and Doucet, A. (2002) Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society* B, **6**4(4), 827-836.

[3] Arulampalam, S., Maskell, S., Gordon, N. and Clapp, T. (2002) A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, **50**(2):174–188.

[4] Bresler, Y. (1986) Two-filter formula for discrete-time non-linear Bayesian smoothing. *International Journal of Control*, **43**(2), 629–641.

[5] Briers, M., Doucet, A. and Maskell, S. (2008) Smoothing algorithms for state-space models, *Annals of the Institute of Statistical Mathematics*, to appear.

[6] Briers, M., Doucet, A. and Singh, S.S. (2005) Sequential auxiliary particle belief propagation, *Proceedings Conference Fusion*.

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

[8] Cappé, O. , Godsill, S. J. and Moulines, E. (2007) An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proceedings*, **95**(5):899–924.

[9] Chopin, N. (2004) Central limit theorem for sequential Monte Carlo and its application to Bayesian inference. *Ann. Statist.*, **32**, 2385-2411.

[10] Crisan, D. and Doucet, A. (2002) A survey of convergence results on particle filtering for practitioners. *IEEE Transactions on Signal Processing*, **50**(3), 736–746.

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

[12] Del Moral, P., Doucet, A. and Jasra, A. (2008) On adaptive resampling procedures for sequential Monte Carlo methods. Technical report. INRIA.

[13] Douc, R., Cappé, O. and Moulines, E. (2005) Comparison of resampling schemes for particle filtering. In 4th International Symposium on Image and Signal Processing and Analysis (ISPA).

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

[15] Doucet, A., Briers, M. and Senecal, S. (2006) Efficient block sampling strategies for sequential Monte Carlo. *Journal of Computational and Graphical Statistics*, **15**(3), 693–711.

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

[17] Doucet, A., de Freitas, N. and Gordon, N.J. (2001) An introduction to sequential Monte Carlo methods. in [16], 3–13.

[18] Fearnhead, P. (2008) Computational methods for complex stochastic systems: A review of some alternatives to MCMC. *Statistics and Computing*, **18**, 151–171.

[19] Fearnhead, P., Wyncoll, D. and Tawn, J. (2008) A sequential smoothing algorithm with linear computational cost. Technical report, Department of Mathematics and Statistics, Lancaster University.

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

[21] Godsill, S.J. and Clapp, T. (2001) Improvement strategies for Monte Carlo particle filters, in [16], 139-158.

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

[23] Johansen, A.M. and Doucet, A. (2008) A note on auxiliary particle filters. *Statistics and Probability Letters*, **78**(12), 1498–1504.

[24] Johansen, A.M. (2008) SMCTC: Sequential Monte Carlo in C++. Technical Report 08-16. University of Bristol, Department of Statistics.

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

[26] Kitagawa, G. and Sato, S. (2001) Monte Carlo smoothing and self-organizing state-space model, in [16], 177–195.

[27] Klaas, M. et al. (2005) Fast particle smoothing: if I had a million particles, in *Proceedings International Conference on Machine Learning*.

[28] Kong, A, Liu, J.S. and Wong, W. H. (1994) Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, **89**, 1032–1044.

[29] Künsch, H. R. (2001) State-space and hidden Markov models. in *Complex Stochastic Systems* (eds. O. E. Barndorff-Nielsen, D. R. Cox and C. Kluppelberg), CRC Press, 109–173.

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

[31] Neal, R. M. (2001) Annealed importance sampling. *Statistics and Computing*, **11**, 125–139.

[32] Olsson, J., Cappé, O., Douc., R. and Moulines, E. (2008) Sequential Monte Carlo smoothing with application to parameter estimation in non-linear state space models. *Bernoulli*, **14**(1):155-179.

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

[34] Pitt, M. K. and Shephard, N. (2001) Auxiliary variable based particle filters, in [16], 271–293.

[35] Robert, C. P. and Casella, G. (2004) *Monte Carlo Statistical Methods*. 2nd edn., Springer-Verlag, New York.

[36] Trotter, H. F., and Tukey, J. W. (1956) Conditional Monte Carlo for normal samples, in *Symposium on Monte Carlo Methods* (ed. H. A. Meyer) Wiley, New York, 64–79.

[37] van der Merwe, R., Doucet, A., De Freitas, N. and Wan, E. (2000) The unscented particle filter. in *Proceedings of Neural and Information Processing Systems*.