Chapter 3

Recent Developments in Auxiliary Particle Filtering

Nick Whiteley[1] *and Adam M. Johansen*[2]

3.1 Background

3.1.1 State Space Models

State space models (SSMs; sometimes termed hidden Markov models, particularly in the discrete case) are very popular statistical models for time series. Such models describe the trajectory of some system of interest as an unobserved E-valued Markov chain, known as the *signal process*. Let $X_1 \sim \nu$ and $X_n|(X_{n-1} = x_{n-1}) \sim f(\cdot|x_{n-1})$ denote this process. Indirect observations are available via an *observation* process, $\{Y_n\}_{n\in\mathbb{N}}$. Conditional upon X_n, Y_n is independent of the remainder of the observation and signal processes, with $Y_n|(X_n = x_n) \sim g(\cdot|x_n)$.

For any sequence $\{z_n\}_{n\in\mathbb{N}}$, we write $z_{i:j} = (z_i, z_{i+1}, ..., z_j)$. In numerous applications, we are interested in estimating, recursively in time, an analytically intractable sequence of posterior distributions $\{p(x_{1:n}|y_{1:n})\}_{n\in\mathbb{N}}$, of the form:

$$p(x_{1:n}|y_{1:n}) \propto \nu(x_1)g(y_1|x_1)\prod_{k=2}^{n} f(x_k|x_{k-1})g(y_k|x_k). \tag{3.1}$$

A great deal has been written about inference for SSMs — see Cappé et al. (2005); Doucet et al. (2001); Doucet and Johansen (2009) for example — and their counterparts in continuous time (Bain and Crisan, 2009). *Filtering*, which corresponds to computing $p(x_n|y_{1:n})$ for each n, is a task of particular interest. Estimation problems in a variety of scientific disciplines can naturally be cast as filtering tasks. A canonical example arises in engineering, where the signal process describes the location and intrinsic parameters of a physical object, observations arise from a noisy sensor and the task is to reconstruct the state of the object as accurately as possible, as observations arrive. Other examples arise from the processing of biological, chemical, seismic, audio, video and financial data. In all these cases the SSM provides a flexible and simple framework in which to describe the relationship between a physically interpretable or abstract hidden process and observed data.

This article is concerned with a class of Monte Carlo algorithms which address the problem of filtering in SSMs by approximating the distributions of interest with a set of weighted random samples. Attention is focused on an algorithm known as the auxiliary particle filter (APF). The APF has seen widespread use in several

[1]Statistics Group, Department of Mathematics, University of Bristol
[2]Department of Statistics, University of Warwick

application areas and a number of algorithms employing the same underlying mechanism have been developed. Existing applications include filtering in object tracking and stochastic volatility models, Pitt and Shephard (1999) (in which the APF was introduced), Carvalho and Lopes (2007); time–varying autoregressive models for audio processing, Andrieu et al. (2003); exact filtering for diffusions, Fearnhead et al. (2008); multi-object tracking, Whiteley et al. (2009) and belief propagation in graphical models Briers et al. (2005).

The remainder of this section introduces a standard approach to the filtering problem, sequential importance resampling (SIR) and describes the APF. Section 3.2 illustrates the strong connection between these algorithms, and provides some guidance upon implementation of the APF. Section 3.3 then illustrates a number of extensions which are suggested by these connections. Section 3.4 describes an elementary technique for variance reduction when applying the APF to SSMs. Termed the stratified APF (sAPF), this algorithm uses low variance sampling mechanisms to assign particles to strata of the state space. The performance of the method is demonstrated in the context of a switching stochastic volatility model using stock index returns data.

3.1.2 Particle filtering

As described above, a common objective when performing inference in SSMs is the recursive approximation of a sequence of posterior distributions (3.1). There are a small number of situations in which these distributions can be obtained in closed form (notably the linear–Gaussian case, which leads to the Kalman filter). However, in general it is necessary to employ approximations. One of the most versatile approaches is to use Sequential Monte Carlo (SMC) methods. Whilst typically more computationally demanding than alternative deterministic techniques (for example see chapter XXXXX), SMC methods are very flexible and have attractive theoretical properties, some of which are discussed below.

The term *particle filtering* is often used to describe the approximation of the optimal filtering equations using SMC techniques. Two common implementations of such algorithms are described in the next two sections. The objective with all such methods is to approximate, sequentially in time, the distribution of X_n given that $Y_{1:n} = y_{1:n}$.

The unnormalised posterior distribution $p(x_{1:n}, y_{1:n})$ given in (3.1) satisfies

$$p(x_{1:n}, y_{1:n}) = p(x_{1:n-1}, y_{1:n-1}) f(x_n|x_{n-1}) g(y_n|x_n). \tag{3.2}$$

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

$$p(x_{1:n}|y_{1:n}) = p(x_{1:n-1}|y_{1:n-1}) \frac{f(x_n|x_{n-1}) g(y_n|x_n)}{p(y_n|y_{1:n-1})}, \tag{3.3}$$

where

$$p(y_n|y_{1:n-1}) = \int p(x_{n-1}|y_{1:n-1}) f(x_n|x_{n-1}) g(y_n|x_n) dx_{n-1:n}. \tag{3.4}$$

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 (3.3)) that

$$p(x_n|y_{1:n}) = \frac{g(y_n|x_n) p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})}, \tag{3.5}$$

where

$$p(x_n|y_{1:n-1}) = \int f(x_n|x_{n-1})p(x_{n-1}|y_{1:n-1})dx_{n-1}. \tag{3.6}$$

Equation (3.6) is known as the prediction step and (3.5) is known as the update step. However, most particle filtering methods rely on a numerical approximation of recursion (3.3) and not of (3.5)-(3.6). This is the case for the majority of the algorithms described in this chapter. One exception, which is described in more detail in section 3.3.1, is the marginal particle filter (Klass et al., 2005) which allows the use of an auxiliary variable technique admitting a similar interpretation to that discussed in the context of the standard APF below.

SMC techniques propagate a collection of weighted samples, termed *particles*, from one iteration to the next in such a way that they provide an approximation of the filtering distribution at each iteration. These collections of particles are used both to approximate integrals with respect to the distributions of interest (and hence to provide estimates of statistics of interest) and to approximate those distributions themselves, thereby allowing inference at the next time step. For a more detailed explanation of these algorithms and an illustration of how most SMC methods may be interpreted as SIR, see Doucet and Johansen (2009).

3.1.3 Sequential Importance Resampling

SIR is one of the simplest SMC approaches to the filtering problem. In fact, as illustrated in algorithm 1, this technique can be used to sample from essentially any sequence of distributions defined on a sequence of spaces of strictly increasing dimension. At its nth iteration, algorithm 1 provides an approximation of $\pi_n(x_{1:n})$. A crucial step in this algorithm is resampling. This involves duplicating particles with high weights, discarding particles with low weights and reweighting to preserve the distribution targeted by the weighted sample. This step prevents a large amount of computational power being wasted on samples with weights close to zero whilst retaining the consistency of associated estimators. The simplest scheme, multinomial resampling, achieves this by drawing N times from the empirical distribution of the weighted particle set (lower variance alternatives are summarised in Doucet et al. (2001) and compared in Douc et al. (2005)).

In a filtering context, $\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n})$ and the expectation of some test function φ_n with respect to the filtering distribution, $\overline{\varphi}_n = \int \varphi_n(x_n)p(x_n|y_{1:n})dx_n$ can be estimated using

$$\widehat{\varphi}^N_{n,SIR} = \sum_{i=1}^{N} W_n^{(i)} \varphi_n(X_n^{(i)})$$

where $W_n^{(i)} = w_n(X^{(i)}_{n-1:n}) \Big/ \sum_{j=1}^N w_n(X^{(j)}_{n-1:n})$ and

$$w_n(x_{n-1:n}) = \frac{\pi_n(x_{1:n})}{q_n(x_n|x_{n-1})\pi_{n-1}(x_{1:n-1})} \propto \frac{g(y_n|x_n)f(x_n|x_{n-1})}{q_n(x_n|x_{n-1})}. \tag{3.7}$$

Note that (3.7) depends upon only the two most recent components of the particle trajectory, and thus algorithm 1 can be implemented with storage requirements which do not increase over time and is suitable for online applications. In fact, SIR can be regarded as a selection-mutation (genetic-type) algorithm constructed with a precise probabilistic interpretation. Viewing SIR as a particle approximation

Algorithm 1 The Generic SIR Algorithm

At time 1
  for $i = 1$ to N **do**
    Sample $X_1^{(i)} \sim q_1(\cdot)$
    Set $W_1^{(i)} \propto \frac{\pi_1(X_1^{(i)})}{q_1(X_1^{(i)})}$
  end for
  Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain $\left\{X_1^{\prime(i)}, \frac{1}{N}\right\}$
At time $n \geq 2$
  for $i = 1$ to N **do**
    Set $X_{1:n-1}^{(i)} = X_{1:n-1}^{\prime(i)}$
    Sample $X_n^{(i)} \sim q_n(\cdot|X_{n-1}^{(i)})$
    Set $W_n^{(i)} \propto \frac{\pi_n(X_{1:n}^{(i)})}{q_n(X_n^{(i)}|X_{n-1}^{(i)})\pi_{n-1}(X_{1:n-1}^{(i)})}$
  end for
  Resample $\left\{X_{1:n}^{(i)}, W_n^{(i)}\right\}$ to obtain $\left\{X_{1:n}^{\prime(i)}, \frac{1}{N}\right\}$

of a Feynman-Kac flow (Del Moral, 2004) allows many theoretical results to be established.

This simple SIR algorithm involves resampling at every iteration. In general, this may not be necessary. Whilst resampling permits stability of the algorithm in the long run, each act of resampling leads to a short term increase in estimator variance. A common strategy, dating back at least to Liu and Chen (1998), is to resample only when the degeneracy of the importance weights, as measured for example by the coefficient of variation, exceeds some threshold. Theoretical analyses of algorithms which resample in this manner have been presented in Del Moral et al. (2008) and Douc and Moulines (2008).

It is commonly accepted that, when designing algorithms for particle filtering, one should endeavour to ensure that the variance of the importance weights is made as small as possible. In pursuit of this objective, it is usual to attempt to employ proposal distributions which are as close as possible to the so-called optimal form, $q_n(x_n|x_{n-1}) \propto f(x_n|x_{n-1})g(y_n|x_n)$ which makes the incremental importance weight independent of x_n. In practice, it is rarely possible to sample from a distribution of the optimal form, although a number of techniques for obtaining good approximations have been developed.

3.1.4 Auxiliary Particle Filters

The use of a well-chosen proposal distribution ensures that knowledge of the current observation is incorporated into the proposal mechanism and so particles are not moved blindly into regions of the state space which are extremely unlikely in light of that observation. However it seems wasteful to resample particles at the end of iteration $n-1$ prior to looking at y_n. That is, it is natural to ask whether it is possible to employ knowledge about the next observation *before* resampling to ensure that particles which are likely to be compatible with that observation have a good chance of surviving — is it possible to preserve diversity in the particle set by taking into account the immediate future as well as the present when carrying out selection? The APF first proposed by Pitt and Shephard (1999, 2001) invoked an auxiliary variable construction in answer to this question.

The essence of this APF was that the sampling step could be modified to sample,

for each particle, an auxiliary variable, corresponding to a particle index, according to a distribution which weights each particle in terms of it compatibility with the coming observation. A suitable weighting is provided by some $\widehat{p}(y_n|x_{n-1})$, an approximation of $p(y_n|x_{n-1}) = \int g(y_n|x_n)f(x_n|x_{n-1})dx_n$ (if the latter is not available analytically). Then the new state value is sampled as the offspring of the particle indicated by this auxiliary variable. It is straightforward to see that this is equivalent to resampling according to those weights before carrying out a standard sampling and resampling iteration. In the terminology of Pitt and Shephard (1999), an APF which employs the exact $p(y_n|x_{n-1})$ and proposes according to $q_n(x_n|x_{n-1}) \propto f(x_n|x_{n-1})g(y_n|x_n)$ is called "fully adapted".

A similar approach in which the auxiliary weights are combined with those of the standard weighting was proposed in Carpenter et al. (1999), which involved a single resampling during each iteration of the algorithm. See algorithm 2.

Algorithm 2 Auxiliary Particle Filter

At time 1
  for $i = 1$ to N **do**
    Sample $X_1^{(i)} \sim q_1(\cdot)$
    Set $\widetilde{W}_1^{(i)} \propto \frac{g(y_1|X_1^{(i)})\nu(X_1^{(i)})}{q_1(X_1^{(i)})}$
  end for
At time $n \geq 2$
  for $i = 1$ to N **do**
    Set $W_{n-1}^{(i)} \propto \widetilde{W}_{n-1}^{(i)} \times \widehat{p}(y_n|X_{n-1}^{(i)})$
  end for
  Resample $\left\{X_{n-1}^{(i)}, W_{n-1}^{(i)}\right\}$ to obtain $\left\{X_{n-1}^{\prime(i)}, \frac{1}{N}\right\}$
  for $i = 1$ to N **do**
    Set $X_{n-1}^{(i)} = X_{n-1}^{\prime(i)}$
    Sample $X_n^{(i)} \sim q_n(\cdot|X_{n-1}^{(i)})$
    Set $\widetilde{W}_n^{(i)} \propto \frac{g(y_n|X_n^{(i)})f(X_n^{(i)}|X_{n-1}^{(i)})}{\widehat{p}(y_n|X_{n-1}^{(i)})q_n(X_n^{(i)}|X_{n-1}^{(i)})}$
  end for

3.2 Interpretation and Implementation

Whilst the APF has seen widespread use, remarkably the first asymptotic analyses of the algorithm have appeared very recently. These analyses provide some significant insights into the performance of the algorithm and emphasise some requirements that a successful implementation must meet.

3.2.1 The APF as SIR

When one considers the APF as a sequence of weighting and sampling operations it becomes apparent that it also has an interpretation as a mutation-selection algorithm. In fact, with a little consideration it is possible to identify the APF as an example of an SIR algorithm.

It was noted in Johansen and Doucet (2008) that the APF described in Carpenter et al. (1999) corresponds to the SIR algorithm which is obtained by setting

$$\pi_n(x_{1:n}) = \widehat{p}(x_{1:n}|y_{1:n+1}) \propto p(x_{1:n}|y_{1:n})\widehat{p}(y_{n+1}|x_n). \tag{3.8}$$

In the SIR interpretation of the APF $p(x_{1:n}|y_{1:n})$ is not approximated directly, but rather importance sampling is used to estimate $\overline{\varphi}_n$, with (weighted) samples which target the importance distribution $\pi_{n-1}(x_{1:n-1})q_n(x_n|x_{n-1})$ provided by an SIR algorithm. The resulting estimate is given by

$$\widehat{\varphi}_{n,APF}^N = \sum_{i=1}^N \widetilde{W}_n^{(i)} \varphi_n(X_n^{(i)}) \tag{3.9}$$

where $\widetilde{W}_n^{(i)} = \widetilde{w}_n(X_{n-1:n}^{(i)}) \Big/ \sum_{j=1}^N \widetilde{w}_n(X_{n-1:n}^{(j)})$ and

$$\widetilde{w}_n(x_{n-1:n}) = \frac{p(x_{1:n}|y_{1:n})}{\pi_{n-1}(x_{1:n-1})q_n(x_n|x_{n-1})} \propto \frac{g(y_n|x_n)f(x_n|x_{n-1})}{\widehat{p}(y_n|x_{n-1})q_n(x_n|x_{n-1})}. \tag{3.10}$$

Note that for a fully adapted APF, the importance weights by which estimation are made are uniform. Only the case in which resampling is carried out once per iteration has been considered here. Empirically this case has been preferred for many years and one would intuitively expect it to lead to lower variance estimates. However, it would be straightforward to apply the same reasoning to the scenario in which resampling is carried out both before and after auxiliary weighting as in the original implementations (doing this leads to an SIR algorithm with twice as many distributions as previously but there is no difficulty in constructing such an algorithm).

Theoretical Consequences

One of the principle advantages of identifying the APF as a particular type of SIR algorithm is that many detailed theoretical results are available for the latter class of algorithm. Indeed, many of the results provided in Del Moral (2004), for example, can be applied directly to the APF via this interpretation. Thus formal convergence results can be obtained without any additional analysis. Via this route, a central limit theorem (CLT), for example, was shown to hold in the case of the APF by Johansen and Doucet (2008). Douc et al. (2009) independently established a CLT for the APF by other means. The asymptotic variance can be decomposed in the same form for it and a simple SIR filter:

Proposition. Under the regularity conditions given in (Chopin, 2004, Theorem 1) or (Del Moral, 2004, Section 9.4, pp. 300-306), which prove this result for SIR algorithms (analysis of SIR and other algorithms can also be found in Douc and Moulines (2008)), we have

$$\sqrt{N}\left(\widehat{\varphi}_{n,SIR}^N - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_{SIR}^2\left(\varphi_n\right)\right),$$
$$\sqrt{N}\left(\widehat{\varphi}_{n,APF}^N - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_{APF}^2\left(\varphi_n\right)\right),$$

where '$\Rightarrow$' denotes convergence in distribution and $\mathcal{N}\left(0, \sigma^2\right)$ is the zero-mean normal of variance σ^2. Moreover, at time $n = 1$ we have

$$\sigma_{SIR}^2\left(\varphi_1\right) = \sigma_{APF}^2\left(\varphi_1\right) = \int \frac{p\left(x_1\right| y_1\right)^2}{q_1\left(x_1\right)} \left(\varphi_1\left(x_1\right) - \overline{\varphi}_1\right)^2 dx_1$$

whereas for $n > 1$

$$\begin{aligned}\sigma^2_{SIR}(\varphi_n) = &\int \frac{p(x_1|y_{1:n})^2}{q_1(x_1)}\Delta\varphi_{1,n}(x_1)^2 dx_1 \\ &+ \sum_{k=2}^{n-1}\int \frac{p(x_{1:k}|y_{1:n})^2}{p(x_{1:k-1}|y_{1:k-1})\,q_k(x_k|x_{k-1})}\Delta\varphi_{k,n}(x_{1:k})^2 dx_{1:k} \\ &+ \int \frac{p(x_{1:n}|y_{1:n})^2}{p(x_{1:n-1}|y_{1:n-1})\,q_n(x_n|x_{n-1})}\left(\varphi_n(x_{1:n}) - \overline{\varphi}_n\right)^2 dx_{1:n},\end{aligned} \tag{3.11}$$

where

$$\Delta\varphi_{k,n}(x_{1:k}) = \int \varphi_n(x_{1:n})p(x_{k+1:n}|y_{k+1:n}, x_k)dx_{k+1:n} - \overline{\varphi}_n$$

and

$$\begin{aligned}\sigma^2_{APF}(\varphi_n) = &\int \frac{p(x_1|y_{1:t})^2}{q_1(x_1)}\Delta\varphi_{1,n}(x_1)^2 dx_1 \\ &+ \sum_{k=2}^{n-1}\int \frac{p(x_{1:k}|y_{1:n})^2}{\widehat{p}(x_{1:k-1}|y_{1:k})q_k(x_k|x_{k-1})}\Delta\varphi_{k,n}(x_{1:k})^2 dx_{1:k} \\ &+ \int \frac{p(x_{1:n}|y_{1:n})^2}{\widehat{p}(x_{1:n-1}|y_{1:n})q_n(x_n|x_{n-1})}\left(\varphi_n(x_{1:n}) - \bar{\varphi}_n\right)^2 dx_{1:n}.\end{aligned} \tag{3.12}$$

Obtaining asymptotic variance expressions in the same form for SIR and the APF allows their comparison on a term-by-term basis. This permits some insight into their relative performance in simple scenarios such as that considered in the following section.

It should, of course, be noted that with a slight change to the conditions (to account for the fact that using the APF one must importance correct estimates relative to those provided by the SIR algorithm which it corresponds to — that is, one integrates a function $\widetilde{w}_n \times \varphi_n$ with respect to an SIR algorithm targeting the auxiliary distributions in order to approximate the expectation of φ_n) essentially any of the results obtained for SIR algorithms can be applied to auxiliary particle filters in the same way. Lastly, we note that by choosing $\widehat{p}(y_n|x_{n-1}) = 1$, one recovers from the APF the SIR algorithm.

3.2.2 Implications for Implementation

It is immediately apparent that, as SIR and the APF are essentially the same algorithm with a different choice of importance weights (in that the the only difference in their implementation is which importance weights are used for resampling and which for estimation) very little additional implementation effort is required to develop both variants of an algorithm). Implementation can be simplified further by employing a generic SMC library such as Johansen (2009).

In real-world settings this may be worthwhile as it may not be straightforward to assess, theoretically, which will provide better estimates at a given computational cost (even when the APF does allow significant reductions in variance to be obtained, it may incur a considerable per-sample cost in the evaluation of a complicated approximation to the predictive likelihood).

From an implementation point of view, perhaps the most significant feature of this interpretation is that it makes clear the criticality of choosing a $\widehat{p}(y_n|x_{n-1})$

which is *more* diffuse than $p(y_n|x_{n-1})$ (as a function of x_{n-1}). For importance sampling schemes in general, it is well known that a proposal distribution with lighter tails than the target distribution can lead to an estimator with infinite variance. In the case of the APF the proposal distribution is defined in terms of $\widehat{p}(y_n|x_{n-1})$, with the importance weight according to which estimates are made being (3.10). It is therefore clear that the popular choice of approximating the predictive likelihood by the likelihood evaluated at the mode of the transition density is a dangerous strategy. This is likely to explain the poor-performance of APF algorithms based on this idea which have appeared in the literature.

A number of other generic approaches lead to more conservative implementations. Each of these techniques may be applicable in some circumstances:

- One simple option is to take

$$\widehat{p}(y_n|x_{n-1}) \propto \int \widehat{g}(y_n|x_n)\widehat{f}(x_n|x_{n-1})dx_n$$

  with the approximations to the likelihood and transition densities being chosen to have heavier tails than the true densities and to permit this integral to be evaluated. For some models it is possible to compute the moments of $p(x_n, y_n|x_{n-1}) = g(y_n|x_n)f(x_n|x_{n-1})$ up to second order, conditional on x_{n-1} Saha et al. (2009). These can then be used to form a Gaussian approximation of $p(x_n, y_n|x_{n-1})$ and thus to $p(y_n|x_{n-1})$, with the variance adjusted to ensure (3.10) is bounded.

- The multivariate t distribution provides a flexible family of approximating distributions: approximating $p(y_n|x_{n-1})$ with a t distribution centred at the mode but with heavier tails than the true predictive likelihood provides a safeguard against excessive concentration whilst remaining similar in spirit to the simple point-approximation approach.

- In cases in which the underlying dynamic model is ergodic, the tractability of the multivariate t distribution provides another strategy. If one approximates the joint distribution of (X_{n-1}, X_n, Y_n) at stationarity with a multivariate t distribution of approximately the correct mean and correlation with tails at least as heavy as those of the true distribution, then one can obtain the marginal distribution of (X_{n-1}, Y_n) under this approximation analytically — it, too, is a multivariate t distribution. Given a multivariate t distribution for (X_{n-1}, Y_n), the conditional density (again, under this approximation) of y_n given x_{n-1} is available in closed form (Nadarajah and Kotz, 2005).

- In the multimodal case, the situation is more complicated. It may be possible to employ a mixture of multivariate t distributions in order to approximate complicated distributions. In very complex settings it may not be practical to approximate the predictive likelihood accurately.

Whilst it remains sensible to attempt to approximate the optimal (in the sense of minimising the variance of the importance weights) transition density

$$q_n(x_n|x_{n-1}) \propto g(y_n|x_n)f(x_n|x_{n-1})$$

and the true predictive likelihood, it is not the case that the APF necessarily outperforms the SIR algorithm using the same proposal even in this setting. This phenomenon is related to the fact that the mechanism by which samples are proposed at the current iteration of the algorithm impacts the variance of estimates made at subsequent time steps.

There are two issues to consider when assessing the asymptotic variance of estimators provided by SIR or APF-type algorithms. Firstly, as the operation performed in both cases is essentially importance sampling, there are likely to be particular functions which are more accurately estimated by each of the algorithms (especially if only a few time steps are considered). An illustrative example was provided Johansen and Doucet (2008) which we discuss in more detail below. The other issue is that the APF can only be expected to provide better estimates in general if, for $k < n$, $p(x_{1:k}|y_{1:k+1})$ is closer to $p(x_{1:k}|y_{1:n})$ than $p(x_{1:k}|y_{1:k})$ is (consider the "importance-weight" terms in the variance decompositions (3.11) and (3.12) in the case where the true predictive likelihood is used). This seems likely to be true for the vast majority of SSMs encountered in practice and so the APF is likely to yield more stable estimates provided that a good approximation of the predictive likelihood is available.

Analysis of Binary State Space Model

In this section we consider the application of SIR and the APF to a very simple SSM. The performance of the two algorithms is then compared in terms of asymptotic variance. The simplicity of the model is such that the asymptotic variances can be evaluated easily in terms of the model parameters, which in this case directly specify the forgetting properties of the signal process and the amount of information provided by the observations. The hope is that, by considering such a simple model, it is possible to gain some insight into the relative performance of SIR and the APF.

The SSM is specified as follows:

$$\begin{array}{lll} E = \{0,1\} & p(x_1 = 0) = 0.5 & p(x_n = x_{n-1}) = 1 - \delta \\ y_n \in E & p(y_n = x_n) = 1 - \varepsilon. & \end{array}$$

The test function $\varphi(x_{1:2}) := x_2$ was used, in the "full adaptation" setting, with $y_{1:2} = (0,1)$:

$$\begin{aligned} q_1(x_1) &:= p(x_1|y_1) & q_n(x_n|x_{n-1}) &:= p(x_n|x_{n-1}, y_n) \\ \widehat{p}(y_n|x_n) &:= \int g(y_n|x_n) f(x_n|x_{n-1}) \mathrm{d}x_n. & & \end{aligned}$$

Figure 3.1(a) illustrates the difference in variance of these two methods as obtained in Johansen and Doucet (2008). In order to understand this, it's useful to consider the asymptotic variance of the two estimators (which follow directly from 3.11 and 3.12):

$$\begin{aligned} \sigma^2_{SIR}(\varphi) = &\int \frac{p(x_1|y_{1:2})^2}{q_1(x_1)} \left(\int \varphi(x_{1:2}) p(x_2|y_2, x_1) dx_2 - \overline{\varphi} \right)^2 dx_1 \\ &+ \int \frac{p(x_{1:2}|y_{1:2})^2}{p(x_1|y_1) q_2(x_2|x_1)} (\varphi(x_{1:2}) - \overline{\varphi})^2 dx_{1:2}, \end{aligned}$$

$$\begin{aligned} \sigma^2_{APF}(\varphi) = &\int \frac{p(x_1|y_{1:2})^2}{q_1(x_1)} \left(\int \varphi(x_{1:2}) p(x_2|y_2, x_1) dx_2 - \overline{\varphi} \right)^2 dx_1 \\ &+ \int \frac{p(x_{1:2}|y_{1:2})^2}{\widehat{p}(x_1|y_{1:2}) q_2(x_2|x_1)} (\varphi(x_{1:2}) - \overline{\varphi})^2 dx_{1:2}. \end{aligned}$$

$\downarrow X_1 \vert X_2 \rightarrow$	0	1
0	$\frac{(1-\delta)(1-\epsilon)\epsilon}{2(1-\delta)\epsilon(1-\epsilon)+\delta((1-\epsilon^2)+\epsilon^2)}$	$\frac{\delta(1-\epsilon)^2}{2(1-\delta)\epsilon(1-\epsilon)+\delta((1-\epsilon^2)+\epsilon^2)}$
1	$\frac{\delta\epsilon^2}{2(1-\delta)\epsilon(1-\epsilon)+\delta((1-\epsilon^2)+\epsilon^2)}$	$\frac{(1-\delta)(1-\epsilon)\epsilon}{2(1-\delta)\epsilon(1-\epsilon)+\delta((1-\epsilon^2)+\epsilon^2)}$

Table 3.1: Target distribution (and APF proposal)

$\downarrow X_1 \vert X_2 \rightarrow$	0	1
0	$\frac{(1-\delta)\epsilon(1-\epsilon)}{\epsilon(1-\delta)+\delta(1-\epsilon)}$	$\frac{\delta(1-\epsilon)^2}{\epsilon(1-\delta)+\delta(1-\epsilon)}$
1	$\frac{\delta\epsilon^2}{\delta\epsilon+(1-\delta)(1-\epsilon)}$	$\frac{(1-\delta)\epsilon(1-\epsilon)}{\delta\epsilon+(1-\delta)(1-\epsilon)}$

Table 3.2: SIR proposal

The first terms of these expansions are identical; the difference between the two algorithms is due entirely to the second term. The SIR term corresponds to the variance of a importance sampling estimate of $\overline{\varphi}$ using $p(x_1|y_{1:2})p(x_2|x_1,y_2)$ as an importance distribution and self-normalised weights:

$$\int \frac{p\left(x_{1:2}\middle|y_{1:2}\right)^2}{p\left(x_1\middle|y_1\right)p\left(x_2\middle|x_1,y_2\right)}\left(\varphi\left(x_{1:2}\right)-\overline{\varphi}\right)^2 dx_{1:2}.$$

The APF term corresponds to the variance of φ under the filtering distribution

$$\int \frac{p(x_{1:2}|y_{1:2})^2}{p(x_1|y_{1:2})p(x_2|x_1,y_2)}\left(\varphi(x_{1:2})-\overline{\varphi}\right)^2 dx_{1:2} = \int p(x_{1:2}|y_{1:2})\left(\varphi(x_{1:2})-\overline{\varphi}\right)^2 dx_{1:2}.$$

The latter can be treated equivalently as the variance of a self-normalised importance sampling estimate using the target distribution as a proposal. Therefore we can appeal to existing results on self-normalised importance sampling estimators in order to compare the two algorithms.

It is well known (Geweke, 1989, Theorem 3) that the optimal proposal distribution (in the sense of minimising the variance) for self-normalised importance sampling is $\propto |\varphi(x)-\overline{\varphi}|\pi(x)$ where φ is the function of interest and π is the target distribution. It is immediately apparent that the marginal distribution of X_1 under the APF proposal distribution is optimal for *any* function which depends only upon X_2. Thus the distribution of X_1 in the SIR expression would definitely *increase* the variance of any estimate *if* the distribution of X_2 was the same in both cases. However, the marginal distribution of X_2 in the two proposal distributions is different and there do exist functions for which that provided by the SIR filter leads to a lower variance.

In the case of interest here, i.e. $\varphi(x_{1:2}) = x_2$, we know that the APF has the optimal marginal distribution for X_1 and that the SIR algorithm will produce samples with an inferior distribution for X_1. Therefore, any instances in which the SIR algorithm produces lower variance estimates are due to the distribution of X_2. For simplicity, we consider the marginal distribution of this variable in what follows noting that in the real scenario, the distribution of X_1 will improve the APF's performance.

The joint distribution of X_1, X_2 in the target (and APF proposal is given in table) 3.1 and that for SIR in table 3.2

It aids interpretation to notice that that $\overline{\varphi} = \sum_{x_2} x_2 p(x_2|y_{1:2}) = p(x_2 = 1|y_{1:2})$. Consequently, the optimal proposal distribution, $q_{opt}(x_2) \propto p(x_2|y_{1:2})|\varphi(x_2)-\overline{\varphi}|$ is

uniform over x_2:

$$q_{opt}(0) \propto p(x_2 = 0|y_{1:2})|\varphi(0) - \overline{\varphi}| = (1 - \overline{\varphi})\overline{\varphi}$$
$$q_{opt}(1) \propto p(x_2 = 1|y_{1:2})|\varphi(1) - \overline{\varphi}| = \overline{\varphi}(1 - \overline{\varphi})$$

This tells us that the marginal distribution for x_2 which minimises the variance of the estimate of *this particular integral* will be uniform over its possible values. The APF generally places more mass on the state supported by the observation than the SIR filter. Consequently, the APF only produces a marginal distribution for X_2 closer to this optimal form when the prior would place the majority of it's mass on the state which is not supported by the observation. Even in this setting, the APF can improve things when we obtain unlikely observations, but may increase the variance when the observation agrees with the prior.

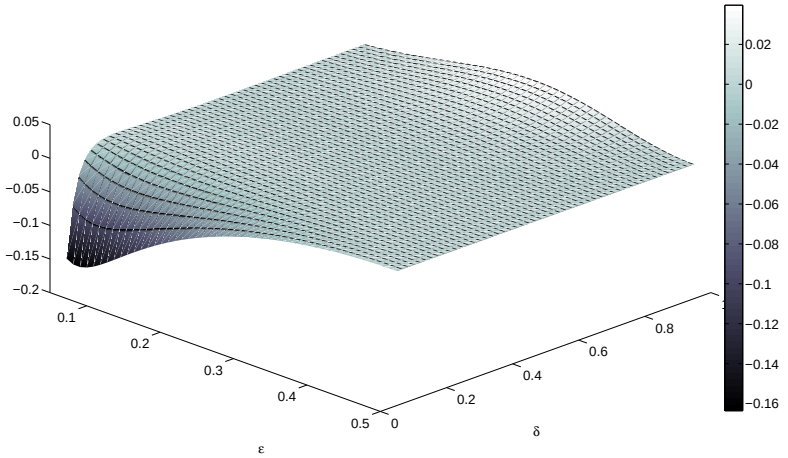
Figure 3.1 illustrates that this mechanism is consistent with what is observed. Figure 3.1(a) shows the difference in estimator variance over a range of values of (δ, ϵ); figures 3.1(b) and 3.1(c) show the marginal probability that $X_2 = 1$ under the proposal distribution associated with the APF and SIR algorithm, respectively and figure 3.1(d) show the difference in $\mathcal{L}_1$ distance to the optimal value for the two approaches. It is clear that the regions in which the SIR algorithm performs well are those in which it provides a much closer to uniform distribution over X_2. Careful inspection reveals that the APF outperforms SIR slightly outside of the regions in which it more closely approximates the uniform distribution over X_2. This is due to the distribution of X_1 (which influences the importance weight) as noted early. It should also be noted that it is when δ and ϵ are both small that one would expect the sub-optimal nature of the SIR distribution over X_1 to have the greatest effect and this is, indeed, where the APF performance is most obviously better.

More generally, one would expect much of the intuition obtained from this simple scenario to apply reasonably directly in more general settings. The APF leads to samples distributed in a way which is closer to the target distribution; it is possible that for some test functions the final step of the APF does not lead to an optimal marginal distribution but this distribution is not intended to operate solely as a device for estimating an integral: it is also used to obtain subsequent distributions and as-such, tracking the sequence of target distributions is of vital importance.

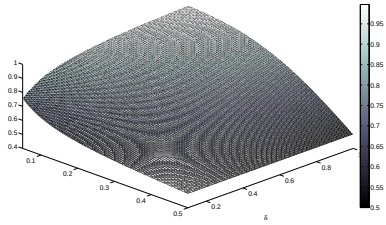
For this reason, minimising incremental variance and otherwise attempting to track these distributions as faithfully as possible remains our preferred method for designing APF algorithms. We also feel that, on average (with respect to observation sequences generated with respect to the true filter, say), the APF is likely to outperform SIR whenever a good approximation to the predictive likelihood is available — especially if a broad class of functions are to be integrated. Note, in particular, the form of the general variance decomposition: it shows that asymptotically, the APF uses distributions of the form $\hat{p}(x_{1:k-1}|y_{1:k})$ to approximate $p(x_{1:k-1}|y_{1:n})$ where the SIR algorithm uses $p(x_{1:k-1}|y_{1:k-1})$. It's approximating the distribution well which will minimise the additional variance which results from these terms and the APF will do this better than SIR (assuming that, at least on average, $p(x_{k-1,k}|y_{1:k})$ is a better proxy for $p(x_{k-1:k}|y_{1:n})$ than $p(x_{k-1}|y_{1:k-1})p(x_k|x_{k-1}, y_k)$ which it will be for any reasonable situation).

3.2.3 Other Interpretations and Developments

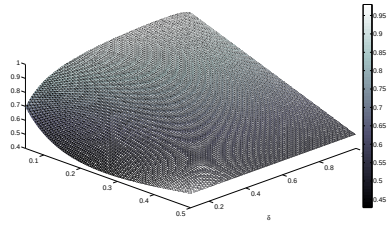
Much has been written about the APF in the decade since it was first proposed.



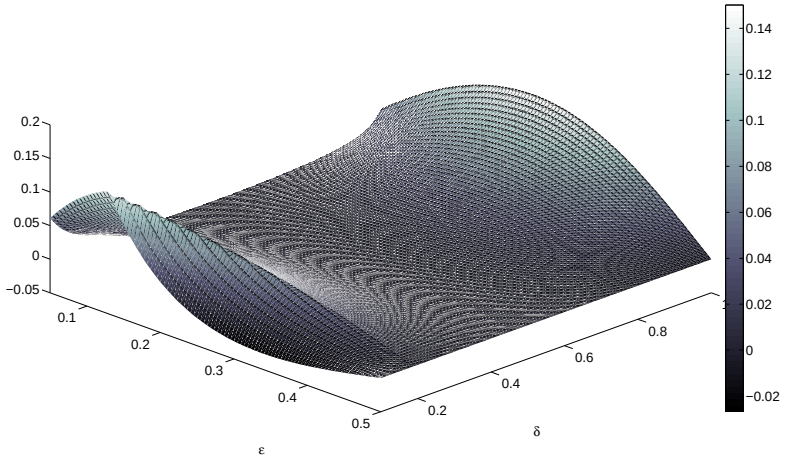
(a) APF Variance - SIR Variance



(b) $\tilde{p}_{\mathrm{APF},2}(X_2 = 1)$



(c) $\tilde{p}_{\mathrm{SIR},2}(X_2 = 1)$



(d) $|\tilde{p}_{\mathrm{APF},2}(X_2 = 1) - 1/2| - |\tilde{p}_{\mathrm{SIR},2}(X_2 = 1) - 1/2|$

Figure 3.1: Properties of the APF and SIR in the binary, perfect-adaptation setting.

Some work based upon the similarity between it and other algorithms precedes that described above. One of the first works to discuss the connections between SIR and the APF was Godsill and Clapp (2001). Closer in spirit to the unified view presented above was Heine (2005) which showed how a number of algorithms could be interpreted within a common framework. This framework differed slightly from that presented above and one of the principle motivations of that approach was the elimination of an explicit resampling step (which is often viewed as being a rather unnatural operation in the discrete-time setting). This seems to be the first paper to observe that "the APF can be considered as an alternative formulation of the general SIR algorithm or vice versa." However, the slightly less standard formulation employed prevents the easy transferal of results from SIR to the APF which was the primary purpose of Johansen and Doucet (2008).

A direct analysis of the particle system underlying the APF was performed recently (Douc et al., 2009) using results obtained in Douc and Moulines (2008). This confirmed the intuitive and empirical results that resampling once per iteration leads to a lower variance estimate than resampling twice. One principle component of this work was the determination of the auxiliary weighting function which minimises the variance of estimates of a particular test function obtained *one step ahead* of the current iterations. The "second stage weights" of (Douc et al., 2009) specify the auxiliary sequence of distributions associated with the auxiliary particle filter. The form which they suggest is optimal for these weights is the following replacement for $\hat{p}(y_{n+1}|x_n)$:

$$\hat{t}_\varphi(x_n, y_{n+1}) = \sqrt{\int \frac{f(x_{k+1}|x_k)^2 g(y_{k+1}|x_{k+1})^2}{q(x_{k+1}|x_k)} (\varphi_{k+1}(x_{1:k+1}) - \overline{\varphi}_{k+1})^2 dx_{k+1}}.$$

Whilst this is of theoretical interest, it requires the computation of a predictive integral which is likely to be even more difficult than that required to obtain the predictive likelihood. In addition to the practical difficulties, it is not clear that it will always be wise to employ the proposed strategy. When performing any Monte Carlo filtering, the particle set is used for two purposes at each time instant: to approximate integrals of interest and to provide an approximation of the distribution required at the next time step. Using this form of weighting is intended to optimise the estimate of the integral at the next time step. However, it need not lead to a good approximation of the distribution itself. Consequently, one may be left with a poorer approximation to the filtering distribution when this weighting function is used than with simpler approaches based upon matching only the distribution and not particular test functions. In such cases, use of this approach may lead to poorer estimation in the future. It is for precisely the same reason that the use of customised proposal distributions tuned for a specific test function are not generally used in particle filtering and thus a more conservative approach, with less adaptation in the proposal mechanism remains sensible.

In subsequent work, a criterion independent of the functions of interest was employed to develop methods for designing adaptive algorithms based upon the auxiliary particle filter in Cornebise et al. (2008). This strategy seeks to minimise the Kullback-Liebler divergence or χ^2-distance between the proposal and target distributions in an adaptive manner (and is similar in spirit to attempting to get as close to the optimal proposal as possible).

3.3 Applications and Extensions

We argue that the innovation of the APF is essentially that, in sampling from a sequence of distributions using a SIR strategy, it can be advantageous to take account of one-step-ahead knowledge about the distributions of interest (more general information could, in principle, be used but it is not easy to envisage realistic scenarios in which this will be practical). This section summarises some other applications of this principle outside of the standard particle filtering domain in which it has previously been applied.

3.3.1 Marginal Particle Filters

As noted above, most particle filtering methods rely on a numerical approximation of (3.3) and not of (3.5)-(3.6) even when only the final time marginal is of interest. This is due to the difficulty associated with evaluating the integral which appears in (3.6) explicitly. One possible solution to this approach, proposed in Klass et al. (2005), is the approximate these integrals using the particle set itself. Doing this increases the computational cost considerably but allows the algorithm to be defined directly on a smaller space than would otherwise be the case. This is of importance when approximating the derivative of the optimal filter in online parameter estimation and optimal control applications Poyiadjis et al. (2005); Kantas (2009).

It is also possible to implement an auxiliary particle filter variant of the marginal particle filter, taking the following form (the standard marginal particle filter is obtained by setting the auxiliary weighting function $\hat{p}(y_{n+1}|x_n)$ to a constant function):

Algorithm 3 Auxiliary Marginal Particle Filter

At time 1

for $i = 1$ to N **do**

Sample $X_1^{(i)} \sim q(\cdot)$.

Set $\widetilde{W}_1^{(i)} \propto \frac{\nu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q(X_1^{(i)})}$.

end for

At time $n \geq 2$

for $i = 1$ to N **do**

Set $W_{n-1}^{(i)} \propto \widetilde{W}_{n-1}^{(i)} \widehat{p}(y_n | X_{n-1}^{(i)})$.

end for

Resample $\left\{X_{n-1}^{(i)}, W_{n-1}^{(i)}\right\}$ to obtain $\left\{X_{n-1}^{\prime(i)}, \frac{1}{N}\right\}$

for $i = 1$ to N **do**

Sample $X_n^{(i)} \sim q(x_n | y_n, X_{n-1}^{\prime(i)})$.

Set $\widetilde{W}_n^{(i)} \propto \frac{g\left(y_n | X_n^{(i)}\right) \sum_{j=1}^N W_{n-1}^{(j)} f\left(X_n^{(i)} | X_{n-1}^{\prime(j)}\right)}{\sum_{j=1}^N W_{n-1}^{(j)} q\left(X_n^{(i)} \middle| y_n, X_{n-1}^{\prime(j)}\right) \widehat{p}(y_n | X_{n-1}^{\prime(j)})}$.

end for

We have presented this algorithm in a form as close as possible to that of the other algorithms described here. It differs in some details from the original formulation. In particular, we do not assume that the predictive likelihood is obtained by approximating the predictive distribution with an atom at its mode — it is not necessary to do this and as discussed in the context of the APF there are some difficulties which may arise as a result of such an approach. As with the APF, it

is necessary to use an importance correction when using this filter to approximate the filtering distributions.

This approach leads to algorithms with a computational complexity which is $\mathcal{O}(N^2)$ in contrast to most particle filters, which are $\mathcal{O}(N)$ algorithms. This would ordinarily be prohibitive, but it was noted in Klass et al. (2005) that techniques widely used for the approximate solution of N-body problems in computational physics and recently applied to statistical learning (Gray and Moore, 2000) can be applied to this problem for a broad class of likelihood functions, thereby reducing the complexity to $\mathcal{O}(N \log N)$ at the cost of a small (and controllable) approximation.

3.3.2 Sequential Monte Carlo Samplers

SMC Samplers are a class of algorithms for sampling iteratively from a sequence of distributions, denoted by $\{\pi_n(x_n)\}_{n\in\mathbb{N}}$, defined upon a sequence of potentially arbitrary spaces, $\{E_n\}_{n\in\mathbb{N}}$, Del Moral et al. (2006a,b). The approach involves the application of SIR to a cleverly constructed sequence of synthetic distributions which admit the distributions of interest as marginals. It is consequently straightforward to employ the same strategy as that used by the APF — see Johansen and Doucet (2007) which also illustrates that convergence results for this class of algorithms follow directly. In this context it is not always clear that there is a good choice of auxiliary distributions, although it is relatively natural in some settings.

The synthetic distributions, $\{\widetilde{\pi}_n(x_{1:n})\}_{n\in\mathbb{N}}$, employed by standard SMC samplers are defined to be

$$\widetilde{\pi}_n(x_{1:n}) = \pi_n(x_n) \prod_{p=1}^{n-1} L_p\left(x_{p+1}, x_p\right),$$

where $\{L_n\}_{n\in\mathbb{N}}$ is a sequence of 'backward' Markov kernels *from* E_n *into* E_{n-1}. With this structure, an importance sample from $\widetilde{\pi}_n$ is obtained by taking the path $x_{1:n-1}$, a sample from $\widetilde{\pi}_{n-1}$, and extending it with a Markov kernel, K_n, which acts from E_{n-1} into E_n, providing samples from $\widetilde{\pi}_{n-1} \times K_n$ and leading to the importance weight:

$$w_n(x_{n-1:n}) = \frac{\widetilde{\pi}_n(x_{1:n})}{\widetilde{\pi}_{n-1}(x_{1:n-1})K_n(x_{n-1}, x_n)} = \frac{\pi_n(x_n)L_{n-1}(x_n, x_{n-1})}{\pi_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}. \quad (3.13)$$

In many applications, each $\pi_n(x_n)$ can only be evaluated pointwise, up to a normalizing constant and the importance weights defined by (3.13) are normalised in the same manner as in the SIR algorithm. Resampling may then be performed.

The optimal (in the sense of minimising the variance of the asymptotic importance weights if resampling is performed at each iteration) choice of L_{n-1} is

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_{n-1}(x_{n-1})K(x_{n-1}, x_n)}{\int \pi_{n-1}(x'_{n-1})K(x'_{n-1}, x_n)dx'_{n-1}}$$

which produces a sampler equivalent to one defined only on the marginal spaces of interest. In practice, it is not generally possible to use the optimal auxiliary kernels and good approximations to this optimal form are required in order to obtain samplers with good variance properties.

If one wishes to sample from a sequence of distributions $\{\pi_n\}_{n\in\mathbb{N}}$ then an alternative to directly implementing an SMC sampler which targets this sequence of distributions, is to employ an auxiliary sequence of distributions, $\{\mu_n\}_{n\in\mathbb{N}}$ and an

importance sampling correction (with weights $\widetilde{w}_n(x_n) = \pi_n(x_n)/\mu_n(x_n)$) to provide estimates. This is very much in the spirit of the APF. Such a strategy was termed auxiliary SMC (ASMC) in Johansen and Doucet (2007). Like the APF, the objective is to maintain a more diverse particle set by using as much information as possible before, rather than after, resampling.

Resample-Move: Inverting Sampling and Resampling

As has been previously noted (Del Moral et al., 2006b) in a setting in which every iteration shares a common state space, $E_n = E$ and in which an MCMC kernel of invariant distribution π_n is employed as the proposal, making use of the auxiliary kernel:

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

the importance weights are simply $w_n(x_{n-1}, x_n) = \pi_n(x_{n-1})/\pi_{n-1}(x_{n-1})$. In addition to its simplicity, this expression has the interesting property that the weight is independent of the proposed state, x_n.

It is possible to interpret this approach as correcting for the discrepancy between the previous and present distributions entirely by importance weighting with the application of an MCMC kernel of the appropriate distribution simply serving to improve the diversity of the sample. It is intuitively clear that one should apply the importance weighting and resample *before* proposing new states in the interests of maximising sample diversity. This has been observed previously. Indeed doing so leads to algorithms with the same structure as the Resample-Move (RM) particle filtering algorithm (Gilks and Berzuini, 2001). By making the following identifications, it is possible to cast this approach into the form of an ASMC sampler.

$$\begin{aligned}
\mu_n(x_n) &= \pi_{n+1}(x_n) \\
L_{n-1}(x_n, x_{n-1}) &= \frac{\mu_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}{\mu_{n-1}(x_n)} = \frac{\pi_n(x_{n-1})K_n(x_{n-1}, x_n)}{\pi_n(x_n)} \\
w_n(x_{n-1:n}) &= \frac{\mu_n(x_n)}{\mu_{n-1}(x_n)} = \frac{\pi_{n+1}(x_n)}{\pi_n(x_n)} \\
\widetilde{w}_n(x_n) &= \mu_{n-1}(x_n)/\mu_n(x_n) = \pi_n(x_n)/\pi_{n+1}(x_n).
\end{aligned}$$

This formal representation allows existing theoretical results to be applied to both RM and its generalisations.

Filtering Piecewise-Deterministic Processes

The SMC Samplers framework was employed by Whiteley et al. (2009) to provide filtering estimates for a class of continuous-time processes. In addition to providing an example of the class of algorithms which are described above, this approach also illustrates that SMC samplers and their auxiliary counterparts can provide useful extensions of SIR-type algorithms in time-series analysis.

Piecewise-Deterministic Processes (PDPs) are a class of stochastic processes whose sample paths, $\{\zeta_t\}_{t\geq 0}$ evolve deterministically in continuous time between a sequence of random times $\{\tau_j\}_{j\in\mathbb{N}}$, at which the path jumps to new, random values $\{\theta_j\}_{j\in\mathbb{N}}$. The prior law of the (τ_j, θ_j) is typically specified by a Markov kernel with density $f(\theta_{n,j}, \tau_{n,j}|\theta_{n,j-1}, \tau_{n,j-1})$.

Filtering for partially observed PDP models involves computing a sequence of posterior distributions given observations $\{Y_n\}_{n\in\mathbb{N}}$. In object tracking applications,

Godsill et al. (2007), the observations may be related to the PDP trajectory by $Y_n = H(\zeta_{t_n}, U_n)$, where U_n is a noise disturbance, H is some non–linear function and $\{t_n\}_{n\in\mathbb{N}}$ is an increasing sequence of observation times. In financial applications such as the pricing of reinsurance Dassios and Jang (2005) and options Centanni and Minozzo (2006), each Y_n is the restriction to the interval $(t_{n-1}, t_n]$ of a Cox process with conditional intensity $(\zeta_t)_{t\in(t_{n-1},t_n]}$. In general, the observation model is specified by a likelihood function $g(y_n|\zeta_{(t_{n-1},t_n]})$.

The nth posterior $\pi_n(k_n, \theta_{n,0:k_n}, \tau_{n,1;k_n}|y_{1:n})$, is a distribution over

$$E_n = \biguplus_{k=0}^{\infty} \{k\} \times \Theta^{k+1} \times \mathbb{T}_{n,k},$$

where $\Theta \subset \mathbb{R}^d$ is a parameter space, $\mathbb{T}_{n,k} = \{\tau_{n,1:k_n} : 0 \leq \tau_{n,1} < ... < \tau_{n,k_n} \leq t_n\}$ and $\biguplus$ indicates disjoint union. The posterior distribution is specified by

$$\pi_n(k_n, \theta_{n,0:k_n}, \tau_{n,1;k_n}|y_{1:n}) \propto$$
$$\nu(\theta_{n,0})S(t_n, \tau_{n,k_n}) \prod_{j=1}^{k_n} f(\theta_{n,j}, \tau_{n,j}|\theta_{n,j-1}, \tau_{n,j-1}) \prod_{p=1}^{n} g(y_n|\zeta_{(t_{n-1},t_n]}),$$

with the convention $\tau_{n,0} = 0$ and where $S(t_n, \tau_{n,k_n})$ is the survivor function associated with the prior distribution over inter-jump times for the interval $[0, t_n]$. The SMC Samplers framework is applied to approximate the distributions of interest, using a proposal kernel consisting of a mixture of moves which extend each particle from E_{n-1} to E_n by adjusting recent jump-time/parameter pairs and adding new ones. An auxiliary scheme for filtering can be obtained by selecting the auxiliary distribution μ_n to be:

$$\mu_n(k_n, \theta_{n,0:k_n}, \tau_{n,1;k_n}) \propto V_n(\theta_{n,k_n}, \tau_{n,k_n})\pi_n(k_n, \theta_{n,0:k_n}, \tau_{n,1;k_n}|y_{1:n}),$$

where $V_n(\theta_{n,k_n}, \tau_{n,k_n})$ is a non-negative potential function which provides information about y_{n+1}. This can be done by approximating the predictive likelihood in the same manner as in the discrete-time case, although some care is required as there may be one or more jumps between observations and these must be considered when approximating that predictive likelihood. This strategy was seen to perform well in Whiteley et al. (2009).

3.3.3 The Probability Hypothesis Density Filter

An unusual application of ideas from the APF can be found in the area of multiple-object tracking. This is an inference task in which one seeks to estimate, in an online manner, the time-varying number and positions of a collection of hidden objects, given a sequence of noisy observations. What makes this task especially difficult is that it is not known which (if any) of the observations arise from which hidden objects. In many applications, the hidden objects are vehicles and the observations arise from sensor measurements, but many other problems in diverse application areas such as communications engineering, biology, audio and music processing can be cast in the same framework. Some examples are noted in Whiteley et al. (2009). See also section XXXXX of this book.

In this scenario, one option is to represent the collection of hidden objects at a single time step as a spatial Poisson process with some inhomogeneous intensity measure. The intensity measure determines the expected number of objects within any region of the state space. Given this representation, the problem of tracking

a large collection of objects is reduced to the problem of approximating, sequentially in time, this intensity measure. The use of SMC methods to approximate this measure has been suggested several times and an auxiliary-particle-filter-type implementation has recently been developed.

In principle, filtering for a multi-object tracking model involves computing a sequence of distributions with essentially the same form as (3.1). Here, E is $E = \biguplus_{k=0}^{\infty} \mathcal{X}^k$, where $\mathcal{X} \subset \mathbb{R}^d$ is the state-space of an individual object: each $X_n = X_{n,1:k_n}$ comprises a random number, k_n, of points, each in $\mathcal{X}$, and can be regarded as a *spatial point process* see Mahler (2007); Singh et al. (2008); Daley and Vere-Jones (2003) for background theory. We refer the reader to section XXXX of this book for further information, but essential to the discussion below is the following concept. The first moment of the distribution of a point process may be specified in terms of an *intensity function*, $\alpha : E \to \mathbb{R}_+$, so that

$$\mathsf{E}[N(A)] = \int_A \alpha_n(x)\mathrm{d}x, \quad A \in \mathcal{B}(\mathcal{X}),$$

where $N(A)$ is the number of points of X which are in the set A and $\mathcal{B}(\mathcal{X})$ is the Borel σ-algebra on $\mathcal{X}$.

A simple multi-object model has the following structure. The hidden objects present at time $n-1$ each survive to time n with location dependent probability $p_S(x_{n-1})$. The surviving objects each evolve independently according to a Markov kernel with density $f(x_n|x_{n-1})$. New objects appear according to a Poisson process with intensity function $\gamma(x_n)$. Each of the surviving and new objects produces an observation with distribution $g(y|x)$. In addition to these detections, spurious observations, termed "clutter", arise from an independent Poisson process with intensity $\kappa(y)$. The observation set at time n therefore consists of a random number of points, $Y_n = Y_{n,1:M_n}$. Crucially, it is not known which of the points of Y_n arise from hidden objects and which are clutter.

Performing filtering when $E = \biguplus_{k=0}^{\infty} \mathcal{X}^k$ is practically very difficult due to the high and variable dimensionality of this space. The Probability Hypothesis Density (PHD) Filter, Mahler (2003), approximates the optimal filter for this problem by assuming that the state process is a-posteriori Poisson (and hence fully characterized by its first moment) and characterising the intensity of that process.

For the model described above, the PHD filtering scheme yields the following prediction/update recursion for intensity functions

$$\alpha_n(x_n) = \int_{\mathcal{X}} f(x_n|x_{n-1})p_S(x_{n-1})\breve{\alpha}_{n-1}(x_{n-1})\mathrm{d}x_{n-1} + \gamma(x_n), \tag{3.14}$$

$$\breve{\alpha}_n(x_n) = \sum_{r=1}^{m_n} \frac{g(y_{n,r}|x_n)}{\mathcal{Z}_{n,r}}\alpha_n(x_n), \tag{3.15}$$

where for $r = 1, 2, ..., m_n$, $\mathcal{Z}_{n,r} = \int_E g(y_{n,r}|x)\alpha_n(x)dx + \kappa(y_{n,r})$. In this notation, $\alpha_n(x)$ and $\breve{\alpha}_n(x)$ are respectively termed the predicted and updated intensities at time n. The problem is then to compute the recursion (3.14)-(3.15) for a given observation sequence, with estimates of k_n and $x_{n,1:k_n}$ made from characteristics of $\breve{\alpha}_n$. For many models this is intractable, due to the integrals involved and because $\breve{\alpha}_n$ is typically of mixture form with a number of components which is increasing in n. Some degree of approximation is therefore required.

SMC methods may be employed to approximate the sequence of intensity functions $\{\breve{\alpha}_n(x_n)\}_{n\in\mathbb{N}}$, (Zajic and Mahler, 2003; Sidenbladh, 2003; Vo et al., 2005; Johansen et al., 2006; Clark and Bell, 2006). In contrast to the case of particle filters which approximate probability distributions, it is necessary for the collection of

weighted samples used here to characterise the total mass of the intensity function in addition to its form. Akin to the APF, an auxiliary SMC implementation has recently been proposed in Whiteley et al. (2009). Empirical results demonstrate that the PHD recursion is particularly well suited to an auxiliary SMC approach. As in the APF, this involves resampling from a particle set which has been re-weighted by a potential function.

In outline, this approach introduces an extended state space $\mathcal{X}' = \mathcal{X} \cup \{s\}$, where s is an isolated "source" point which does not belong to $\mathcal{X}$, then defines an intensity function denoted $\beta_n(x_{n-1:n}, r)$ on $\mathcal{X} \times \mathcal{X}' \times \{1, ..., m_n\}$ as follows:

$$\beta_n(x_{n-1:n}, r) = \frac{g(y_{n,r}|x_n)}{\mathcal{Z}_{n,r}} \left[f(x_n|x_{n-1}) p_S(x_{n-1}) \breve{\alpha}_{n-1}(x_{n-1}) \mathbb{I}_{\mathcal{X}}(x_{n-1}) + \gamma(x_n) \delta_s(x_{n-1}) \right]. \tag{3.16}$$

Note that

$$\breve{\alpha}_n(x_n) = \sum_{r=1}^{m_n} \int_{\mathcal{X}'} \beta_n(x_{n-1:n}, r) \mathrm{d}x_{n-1}. \tag{3.17}$$

The algorithm of Whiteley et al. (2009) uses IS to target (3.16) and thus yields a particle approximation of $\breve{\alpha}_n(x_n)$ due to (3.17).

Assume that there is available a particle approximation of $\breve{\alpha}_{n-1}(x_{n-1})$, denoted by $\breve{\alpha}^N_{n-1}(x_{n-1})$. N samples are drawn from some distribution $q_n(r)$ over $\{1, 2, ..., m_n\}$, yielding $\{R^{(i)}_n\}_{i=1}^N$. For each i, $X^{(i)}_{n-1}$ is then drawn from

$$\pi^N_{n-1,R^{(i)}}(x_{n-1}) \propto \widehat{p}(y_{n,R^{(i)}}|x_{n-1}) \left[\breve{\alpha}^N_{n-1}(x_{n-1}) \mathbb{I}_{\mathcal{X}}(x_{n-1}) + \delta_s(x_{n-1}) \right], \tag{3.18}$$

$\widehat{p}(y_{n,r}|x_{n-1})$ being an approximation of $p(y_{n,r}|x_{n-1})$, which is itself defined by

$$p(y_{n,r}|x_{n-1}) = \mathbb{I}_{\mathcal{X}}(x_{n-1}) p_S(x_{n-1}) \int_{\mathcal{X}} g(y_{n,r}|x_n) f(x_n|x_{n-1}) dx_n + \mathbb{I}_{\{s\}}(x_{n-1}) \int_{\mathcal{X}} g(y_{n,r}|x_n) \gamma(x_n) dx_n.$$

For each i, $X^{(i)}_n$ is then drawn from a kernel $q_n(\cdot|X^{(i)}_{n-1}, R^{(i)}_n)$. The importance weight which targets (3.16) is given by:

$$\widetilde{w}_n(x_{n-1:n}, r) \propto \frac{g(y_{n,r}|x_n)[f(x_n|x_{n-1}) p_S(x_{n-1}) \mathbb{I}_{\mathcal{X}}(x_{n-1}) + \gamma(x_n) \mathbb{I}_{\{s\}}(x_{n-1})]}{q_{n,r}(x_n|x_{n-1}) \pi_{n-1,r}(x_{n-1}) q_n(r)},$$

with each normalizing constant $\mathcal{Z}_{n,r}$ also estimated by IS, much as in SMC algorithms for SSMs. The result is a particle approximation of $\breve{\alpha}_n(x_n)$. Whiteley et al. (2009) also shows how to choose $q_n(r)$ in an optimal fashion.

The connection with the APF is evident from the form of (3.18): drawing from this auxiliary distribution involves resampling from the existing particle set reweighted by a potential function incorporating knowledge of the next observation. As demonstrated empirically in Whiteley et al. (2009), compared to non-auxiliary SMC implementations, this method can result in importance weights of lower variance and more reliable estimates.

3.4 Further Stratifying the APF

It is common knowledge that the use of multinomial resampling in a particle filter unnecessarily increases the Monte Carlo variance of the associated estimators and that the use of residual, systematic or stratified approaches can significantly reduce that variance (Douc et al., 2005). This is also true in the case of the APF and one should always employ minimum variance resampling strategies. Under some circumstances it may be possible to a achieve a further variance reduction in the APF.

Consider again the SSM from section 3.1.1. Let $(E_j)_{j=1}^M$ denote a partition of E. Introducing an auxiliary stratum-indicator variable, $s_n = \sum_{j=1}^M j\mathbb{I}_{E_j}(x_n)$, we redefine the SSM on a higher dimensional space, with the signal process being $E \times \{1, 2, ..., M\}$-valued, with transition kernel

$$r(x_n, s_n|x_{n-1}, s_{n-1}) = r(x_n|s_n, x_{n-1})r(s_n|x_{n-1}),$$

where

$$r(x_n|s_n, x_{n-1}) \propto \mathbb{I}_{E_{s_n}}(x_n)f(x_n|x_{n-1}), \quad r(s_n|x_{n-1}) = \int_{E_{s_n}} f(x_n|x_{n-1})\mathrm{d}x_n.$$

The initial distribution of the extended chain is defined in a similar manner and the likelihood function remains unchanged. The posterior distributions for the extended model then obey the following recursion:

$$p(x_{1:n}, s_{1:n}|y_{1:n}) \propto g(y_n|x_n)r(x_n|s_n, x_{n-1})r(s_n|x_{n-1})p(x_{1:n-1}, s_{1:n-1}|y_{1:n-1}). \tag{3.19}$$

Note that the marginal distribution of $x_{1:n}$ in (3.19) coincides with the original model.

As in the SIR interpretation of the APF, we then construct an auxiliary sequence of distributions, $\{\pi(x_{1:n-1}, s_{1:n})\}_{n\in\mathbb{N}}$, which will be targeted with an SIR algorithm, where:

$$\pi(x_{1:n-1}, s_{1:n}) \propto \widehat{p}(y_n|s_n, x_{n-1})\widehat{r}(s_n|x_{n-1})p(x_{1:n-1}, s_{1:n-1}|y_{1:n-1}). \tag{3.20}$$

The key feature of (3.20) is that the resampling step of the corresponding SIR algorithm will select pairs of previous state values x_{n-1} and current strata s_n. This assignment can be performed with a low variance resampling mechanism. The corresponding algorithm, which we refer to as the stratified Auxiliary Particle Filter (sAPF) is given below.

For each i, we first draw each $X_n^{(i)}|x_{n-1}^{(i)}, s_n^{(i)} \sim q_n(\cdot|x_{n-1}^{(i)}, s_n^{(i)})$. Then, instead of randomly sampling a value $s_{n+1}^{(i)}$, we evaluate one importance weight for every possible value of s_{n+1}, resulting in a collection of $N \times M$ weighted sample points. The resampling step of the SIR algorithm then draws N times from the resulting distribution on $\{1, 2, ..., N\} \times \{1, 2, ..., M\}$. The method of Karlsson and Bergman (2000), proposed in the context of a particular class of tracking problems can be viewed as a special case of the proposed algorithm. However, Karlsson and Bergman (2000) did not employ low–variance resampling schemes, which is, as will be shown below, the key to obtaining both a variance reduction and a decrease in computational cost.

Algorithm 4 Stratified Auxiliary Particle Filter

At time 1

for $i = 1$ to N **do**

Sample $X_1^{(i)} \sim q_1(\cdot)$

Set $\widetilde{W}_1^{(i)} \propto \frac{g(y_1|X_1^{(i)})\nu(X_1^{(i)})}{q_1(X_1^{(i)})}$

end for

At time $n \geq 2$

for $i = 1$ to N **do**

for $j = 1$ to M **do**

Set $W_{n-1}^{(i,j)} \propto \widetilde{W}_{n-1}^{(i)} \times \widehat{p}(y_n|X_{n-1}^{(i)}, s_n = j)\widehat{r}(s_n = j|X_{n-1}^{(i)})$

end for

end for

Resample $\left\{X_{n-1}^{(i)}, j, W_{n-1}^{(i,j)}\right\}_{(i,j)\in\{1,\ldots,N\}\times\{1,\ldots,M\}}$ to obtain $\left\{X_{n-1}^{\prime(i)}, S_n^{(i)}, \frac{1}{N}\right\}$

for $i = 1$ to N **do**

Set $X_{n-1}^{(i)} = X_{n-1}^{\prime(i)}$

Sample $X_n^{(i)} \sim q_n(\cdot|X_{n-1}^{(i)}, S_n^{(i)})$

Set $\widetilde{W}_n^{(i)} \propto \frac{g(y_n|X_n^{(i)})f(X_n^{(i)}|X_{n-1}^{(i)})}{\widehat{p}(y_n|X_{n-1}^{(i)},S_n^{(i)})\widehat{r}(S_n^{(i)}|X_{n-1}^{(i)})q_n(X_n^{(i)}|X_{n-1}^{(i)},S_n^{(i)})}$

end for

The importance weight which targets $p(x_{1:n}, s_{1:n}|y_{1:n})$ (i.e. the analogue of (3.10)) is then:

$$\widetilde{w}_n(x_{n-1:n}, s_n) \propto \frac{g(y_n|x_n)f(x_n|x_{n-1})}{\widehat{p}(y_n|s_n, x_{n-1})\widehat{r}(s_n|x_{n-1})q_n(x_n|s_n, x_{n-1})}.$$

This effectively assigns both a parent particle *and* a stratum to each offspring. Crucially, this assignment can be performed with a low variance resampling mechanism. This approach is especially of interest in the context of *switching* SSMs, where the state space has a natural partition structure by definition. We consider the application to such models below. First, we consider the effect of the above sampling scheme on the conditional variance of the resulting estimates.

3.4.1 Reduction in Conditional Variance

The following section illustrates the principal benefit of the proposed approach: a significant reduction in computational cost in those situations in which a natural stratification exists. This section illustrates that incorporating this additional stratification cannot make things worse in the sense that the variance of resulting estimators will be at most the same as those obtained with a non-stratified variant of those estimators. For simplicity we compare the performance of the sAPF to that of the APF in terms of the conditional variance arising from a single act of resampling and assigning particles to strata.

There are several ways in which one might go about using a low–variance mechanism in the resampling step of algorithm 4. All the methods described in Douc et al. (2005) are applicable and in this section we consider one way of using the *stratified* resampling mechanism, see Kitagawa (1996); Fearnhead (1998). This method uses a form of inversion sampling to draw N samples the distribution defined by the particle set. Inversion sampling itself involves generating $\mathcal{U}[0, 1]$ random variates and passing them through the generalised inverse of the target distribution function

Robert and Casella (2004). The stratified resampling scheme is so-named because it involves partitioning $[0,1]$ into N strata of length $1/N$. A single uniform variate is then drawn on each sub–interval and passed through the inverse of the CDF, see Douc et al. (2005) for further details.

We next consider how the stratified resampling mechanism could be used in the sAPF and how it would be used in the regular APF. It should be noted that the scheme described below is not the *only* way in which stratified resampling can be applied within the sAPF. Indeed there are alternatives which may be of even lower variance. However, the scheme described below is simple enough to permit direct analysis, providing some insight into how variance reduction can be achieved.

As part of the discussion which follows, we need some notation to indicate when the stratified resampling scheme is used and to specify the resulting random variables.

For a collection of random variables $\{X^{(i)}\}_{i=1}^N$, and a probability distribution μ, we use the notation $\{X^{(i)}\}_{i=1}^N \overset{ss}{\sim} \mu$ to indicate that the samples $\{X^{(i)}\}_{i=1}^N$ are generated using the stratified resampling mechanism targeting μ. Consider a collection of weighted samples $\{X^{(i)}, W^{(i)}\}_{i=1}^N$ such that $\sum_{i=1}^N W^{(i)} = 1$ and the associated empirical probability distribution

$$\sum_{i=1}^N W^{(i)} \delta_{X^{(i)}}(dx).$$

Resampling N times from this distribution can be interpreted as generating, via some mechanism, a set of N ancestors, with $A^{(i)}$ denoting the ancestor of the i^{th} particle so that the resulting empirical distribution can be written as

$$\frac{1}{N}\sum_{i=1}^N \delta_{X^{(A^{(i)})}}(dx),$$

i.e. in relation to the notation of algorithm 4, $X'^{(i)} \equiv X^{(A^{(i)})}$. It will also be convenient to specify the number of replicates of each existing sample and a cumulative count of these replicates, so for $i \in \{1, ..., N\}$ we define

$$N_i = \sum_{j=1}^N \mathbb{I}_{[A^{(j)}=i]}, \qquad N_i^* = \sum_{j=1}^i N_j.$$

Finally, to connect with the notation of algorithm 4 we also set

$$W^{(i)} = \sum_{j=1}^M W^{(i,j)}, \qquad W^{(j|i)} = \frac{W^{(i,j)}}{\sum_{j=1}^M W^{(i,j)}}, \tag{3.21}$$

where the time index has been suppressed (as it is throughout this section) for clarity.

With these conventions in hand, we consider a set, $\{X^{(i)}, W^{(i)}\}_{i=1}^N$, of weighted samples resulting from some iteration of an SMC algorithm and conditional upon this weighted sample set, analyze the variance arising from resampling the particles and assigning particles to strata.

Table 3.3 shows how an algorithm which employs the stratified sampling mechanism in both the resampling and strata-selection steps can be compared with the standard algorithm. Figure 3.2 shows a graphical representation of the procedures.

APF	**sAPF**
$\{A^{(i)}\}_{i=1}^{N} \overset{ss}{\sim} \sum_{i=1}^{N} W^{(i)}\delta_i(da)$	$\{A^{(i)}\}_{i=1}^{N} \overset{ss}{\sim} \sum_{i=1}^{N} W^{(i)}\delta_i(da)$

for $i = 1$ to N
$S^{(i)} \sim \sum_{j=1}^{M} W^{(j\|A^{(i)})}\delta_j(ds)$
end for
	for $i = 1$ to N
	if $N_i > 0$
	$\{S^{(j)}\}_{j=N_{i-1}^{*}+1}^{N_i^{*}} \overset{ss}{\sim} \sum_{j=1}^{M} W^{(j\|i)}\delta_j(ds)$
	end if
	end for

Table 3.3: Resampling and assignment to strata for the APF and sAPF algorithms, both employing stratified sampling. Here, the APF uses the low-variance sampling mechanism in assigning ancestors. By contrast, the sAPF uses the low-variance mechanism in both assiging ancestors and strata.

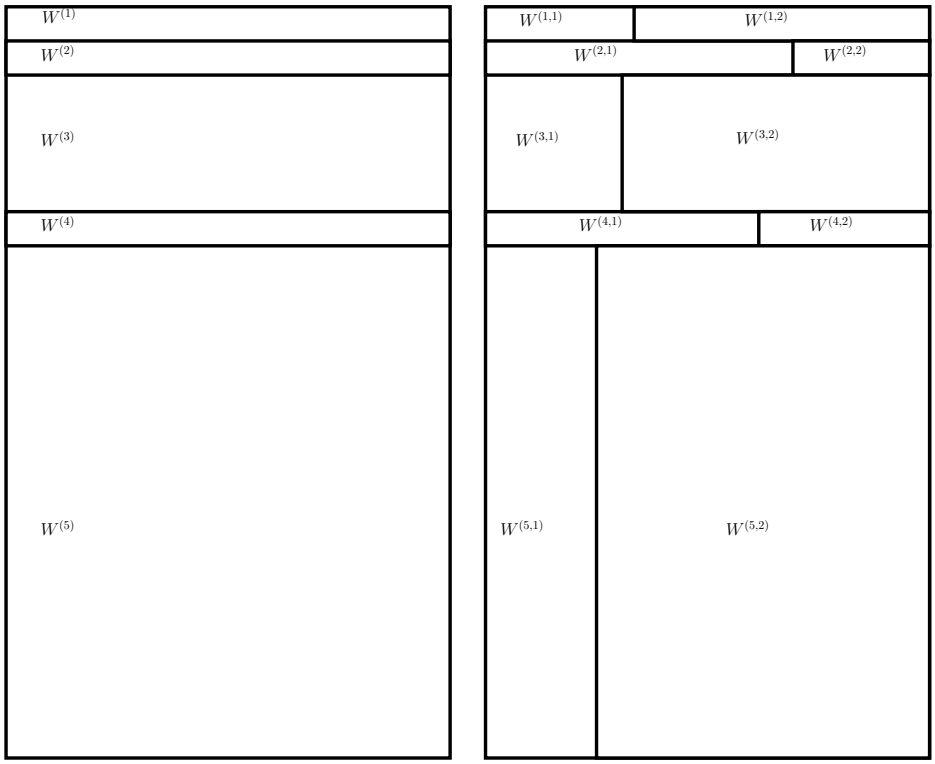


Figure 3.2: An illustration of stratified resampling within the APF (left) and the sAPF (right) with $N = 5$ particles and $M = 2$ strata. For the APF, each box corresponds to an existing particle; for the sAPF, each box corresponds to an existing particle/stratum pair. In both cases, the area of each box is proportional to the corresponding weight and a number of particles proportional to the area of each box is sampled with the appropriate parameters. In the case of the APF the boxes have heights proportional to the weights of the particles and constant width: only the parent particle is assigned by the low–variance sampling mechanism. In the case of the sAPF the height of the boxes remains proportional to the weight of the particle, $W^{(i)} = \sum_j W^{(i,j)}$, but now the assignment of both parent and stratum is performed using the low–variance sampling mechanism.

Given the weighted sample set $\{X^{(i)}, W^{(i)}\}_{i=1}^{N}$, procedures of table 3.3 both

result in a set of ancestor and strata indicators. For a function $\varphi : \{1, ..., M\} \times E \to \mathbb{R}$ we write $\widehat{\varphi}^N_{\text{sAPF}}$ and $\widehat{\varphi}^N_{\text{APF}}$ for the estimators of the form

$$\frac{1}{N}\sum_{i=1}^{N} \varphi(S^{(i)}, X^{(A^{(i)})})$$

which arise from the sAPF and the APF, respectively. The following proposition establishes that the sAPF scheme of table 3.3 does indeed yield a reduction in conditional variance over the APF scheme.

Proposition 1. *For an integrable function $\varphi : \{1, ..., M\} \times E \to \mathbb{R}$ and for all N,*

$$\mathsf{V}(\widehat{\varphi}^N_{sAPF}|\mathcal{F}) \leq \mathsf{V}(\widehat{\varphi}^N_{APF}|\mathcal{F}).$$

where $\mathcal{F} = \sigma(\{X^{(i)}, W^{(i)}\}_{i=1}^N)$.

Proof. The variances are first decomposed in the following manner

$$\mathsf{V}(\widehat{\varphi}^N_{\text{sAPF}}|\mathcal{F}) = \mathsf{E}(\mathsf{V}(\widehat{\varphi}^N_{\text{sAPF}}|\mathcal{G})|\mathcal{F}) + \mathsf{V}(\mathsf{E}(\widehat{\varphi}^N_{\text{sAPF}}|\mathcal{G})|\mathcal{F}) \tag{3.22}$$

$$\mathsf{V}(\widehat{\varphi}^N_{\text{APF}}|\mathcal{F}) = \mathsf{E}(\mathsf{V}(\widehat{\varphi}^N_{\text{APF}}|\mathcal{G})|\mathcal{F}) + \mathsf{V}(\mathsf{E}(\widehat{\varphi}^N_{\text{APF}}|\mathcal{G})|\mathcal{F}), \tag{3.23}$$

where $\mathcal{G} = \mathcal{F} \vee \sigma(\{A^{(i)}\}_{i=1}^N)$. Comparison is then performed term-by-term. First consider the conditional expectations:

$$\begin{aligned}\mathsf{E}(\widehat{\varphi}^N_{\text{sAPF}}|\mathcal{G}) &= \frac{1}{N}\sum_{\{i:N_i>0\}}\sum_{j=1}^{N_i}\int_{\frac{j-1}{N_i}}^{\frac{j}{N_i}} N_i\varphi(D_i^{\text{inv}}(u), X^{(i)})\mathrm{d}u \\ &= \frac{1}{N}\sum_{\{i:N_i>0\}}^{N} N_i\int_0^1 \varphi(D_i^{\text{inv}}(u), X^{(i)})\mathrm{d}u = \mathsf{E}(\widehat{\varphi}^N_{\text{APF}}|\mathcal{G}).\end{aligned} \tag{3.24}$$

where D_i^{inv} is the generalised inverse CDF associated with $\sum_{j=1}^M W^{(j|i)}\delta_j$.

Next consider the conditional variances. First note that for both the sAPF and APF the $\{S^{(i)}\}_{i=1}^N$ are conditionally independent given $\{A^{(i)}\}_{i=1}^N$. Hence:

$$\begin{aligned}\mathsf{V}(\widehat{\varphi}^N_{\text{sAPF}}|\mathcal{G}) =& \frac{1}{N^2}\sum_{i=1}^{N}\mathsf{E}([\varphi(S^{(i)}, X^{(A^{(i)})})]^2|\mathcal{G}) \\ &- \frac{1}{N^2}\sum_{i=1}^{N}[\mathsf{E}(\varphi(S^{(i)}, X^{(A^{(i)})})|\mathcal{G})]^2 \\ =& \frac{1}{N^2}\sum_{i=1}^{N} N_i\int_0^1 [\varphi(D_i^{\text{inv}}(u), X^{(i)})]^2\mathrm{d}u \\ &- \frac{1}{N^2}\sum_{\{i:N_i>0\}}\sum_{j=1}^{N_i}\left[\int_{\frac{j-1}{N_i}}^{\frac{j}{N_i}} N_i\varphi(D_i^{\text{inv}}(u), X^{(i)})\mathrm{d}u\right]^2,\end{aligned} \tag{3.25}$$

whereas for the APF,

$$\begin{aligned}\mathsf{V}(\widehat{\varphi}^N_{\text{APF}}|\mathcal{G}) =& \frac{1}{N^2}\sum_{i=1}^{N} N_i\int_0^1 [\varphi(D_i^{\text{inv}}(u), X^{(i)})]^2\mathrm{d}u \\ &- \frac{1}{N^2}\sum_{i=1}^{N} N_i\left[\int_0^1 \varphi(D_i^{\text{inv}}(u), X^{(i)})\mathrm{d}u\right]^2.\end{aligned} \tag{3.26}$$

Applying Jensen's inequality to the second term in (3.25) and (3.26) shows that

$$\mathsf{V}(\widehat{\varphi}^N_{\mathrm{sAPF}}|\mathcal{G}) \leq \mathsf{V}(\widehat{\varphi}^N_{\mathrm{APF}}|\mathcal{G}). \tag{3.27}$$

The result follows upon combining (3.22), (3.23), (3.24) and (3.27). □

It is stressed that Proposition 1 deals with the *conditional* variance, given $\{X^{(i)}, W^{(i)}\}_{i=1}^N$. This gives some insight into the performance of the algorithm, but ideally one would like to confirm a reduction in the unconditional variance. In the case that residual resampling is used it may be possible to apply similar techniques to those used in Chopin (2004) in order to establish a reduction in unconditional asymptotic variance.

3.4.2 Application to Switching State Space Models

Switching SSMs are a particular class of models in which the state of the unobserved process can be expressed in terms of two components, $X_n = (S_n, \theta_n)$, with s_n valued in $\{1, 2, ..., M\}$ and θ_n valued in some space Θ, typically a subset of $\mathbb{R}^d$. The corresponding state space is of the form

$$E = \{1, \ldots, M\} \times \Theta = \biguplus_{j=1}^{M} \{j\} \times \Theta,$$

so the state space has a natural partition structure. Note that $E_j = \{j\} \times \Theta$ so automatically we have $s_n = \sum_{j=1}^M j \mathbb{I}_{E_j}(x_n)$ as before.

We will focus on models of the form:

$$p(\theta_{1:n}, s_{1:n}|y_{1:n}) \propto g(y_n|\theta_n) r(\theta_n|\theta_{n-1}, s_n) r(s_n|s_{n-1}) p(\theta_{1:n-1}, s_{1:n-1}|y_{1:n-1}), \tag{3.28}$$

which arise in a wide variety of applications, including target tracking, Doucet et al. (2001); audio signal processing, Andrieu et al. (2003); and econometrics Carvalho and Lopes (2007). Note that due to the structure of the model we have $r(s_n|x_{n-1}) = r(s_n|s_{n-1})$.

In this model s_n is a latent state, which is not observed. The model of the hidden process $(\theta_n)_{n\in\mathbb{N}}$ can be interpreted as *switching* between M distinct dynamic regimes, with transitions between these regimes governed a-priori by the transition kernel $r(s_n|s_{n-1})$. This allows a larger degree of flexibility than in standard SSMs and is especially useful for modelling time–series which exhibit temporal heterogeneity.

In the conditionally linear–Gaussian case, given a trajectory $s_{1:n}$ it is possible to compute $p(\theta_{1:n}|y_{1:n}, s_{1:n})$ using the Kalman filter and thus SMC algorithms for filtering can be devised in which the θ components of the state are integrated out analytically, see Doucet et al. (2001). We do not assume such structure, although the methods described above are applicable in that case. The sAPF algorithm for the specific case of switching state space models is given below.

We next consider application of the sAPF to a Markov–switching stochastic volatility (SV) model, as studied in Carvalho and Lopes (2007). SV models with switching regime allow occasional discrete shifts in the parameter determining the level of the log volatility of financial returns. They have been advocated as a means by which to avoid overestimation of volatility persistence, see So et al. (1998) and references therein.

Algorithm 5 sAPF for Switching State Space Models

At time 1

for $i = 1$ to N **do**

Sample $(\theta_1^{(i)}, S_1^{(i)}) \sim q_1(\cdot)$

Set $\widetilde{W}_1^{(i)} \propto \frac{g(y_1|\theta_1^{(i)})\nu(\theta_1^{(i)}, S_1^{(i)})}{q_1(\theta_1^{(i)}, S_1^{(i)})}$

end for

At time $n \geq 2$

for $i = 1$ to N **do**

for $j = 1$ to M **do**

Set $W_{n-1}^{(i,j)} \propto \widetilde{W}_{n-1}^{(i)} \times \widehat{p}(y_n|\theta_{n-1}^{(i)}, s_n = j) r(s_n = j|S_{n-1}^{(i)})$

end for

end for

Resample $\left\{\theta_{n-1}^{(i)}, j, W_{n-1}^{(i,j)}\right\}_{(i,j)\in\{1,\ldots,N\}\times\{1,\ldots,M\}}$ to obtain $\left\{\theta_{n-1}^{\prime(i)}, S_n^{(i)}, \frac{1}{N}\right\}$

for $i = 1$ to N **do**

Set $\theta_{n-1}^{(i)} = \theta_{n-1}^{\prime(i)}$

Sample $\theta_n^{(i)} \sim q_n(\cdot|\theta_{n-1}^{(i)}, S_n^{(i)})$

Set $\widetilde{W}_n^{(i)} \propto \frac{g(y_n|\theta_n^{(i)}) r(\theta_n^{(i)}|\theta_{n-1}^{(i)}, S_n^{(i)})}{\widehat{p}(y_n|\theta_{n-1}^{(i)}, S_n^{(i)}) q_n(\theta_n^{(i)}|\theta_{n-1}^{(i)}, S_n^{(i)})}$

end for

The log–volatility process $\{\theta_n\}_{n\in\mathbb{N}}$ and observations $\{Y_n\}_{n\in\mathbb{N}}$ obey the following equations

$$\theta_n = \phi\theta_{n-1} + \alpha_{s_n} + \zeta_n,$$
$$Y_n = \epsilon_n \exp(\theta_n/2),$$

where ζ_n is an independent $\mathcal{N}(0, \sigma_\theta^2)$ random variable and ϵ_n is an independent $\mathcal{N}(0,1)$ random variable. The parameter ϕ is the persistence of volatility shocks, $\{\alpha_j\}_{j=1}^M$ are the log-volatility levels and s_n is the latent regime indicator so that

$$\alpha_{s_n} = \gamma_1 + \sum_{j=2}^{M} \gamma_j \mathbf{I}_{[s_n \geq j]},$$

where $\{\gamma_j\}_{j=1}^M$ are log–volatility increments. The prior transition kernel $r(s_n|s_{n-1})$ is specified by a stochastic matrix with entry p_{kl} being the probability of a transition from state k to state l.

In order to construct the potential function $\widehat{p}(y_n|\theta_{n-1}, s_n)$ and the proposal distribution $q_n(\theta_n|\theta_{n-1}, s_n)$ we employ a slight modification of the technique proposed in Pitt and Shephard (1999) for standard SV models. The idea is to exploit the log–concavity of the likelihood function and form an approximation of $g(y_n|\theta_n)$ by taking a first order Taylor expansion of the log–likelihood about the conditional mean of θ_n. With an abuse of notation we write $\bar{\theta}_n := \phi\theta_{n-1} + \alpha_{s_n}$. The approximation of the likelihood is then specified by

$$\log \widehat{g}(y_n|\theta_n; \theta_{n-1}, s_n) = \log g(y_n|\bar{\theta}_n) + (\theta_n - \bar{\theta}_n) \cdot \frac{\partial}{\partial\theta} \log g(y_n|\theta)\Big|_{\bar{\theta}_n}. \tag{3.29}$$

We then choose

$$q_n(\theta_n|s_n,\theta_{n-1}) \propto \widehat{g}(y_n|\theta_n;\theta_{n-1},s_n)r(\theta_n|\theta_{n-1},s_n),$$

which is a Gaussian density, $\mathcal{N}(\mu_{q_n},\sigma^2_{q_n})$, with parameters

$$\mu_{q_n} = \phi\theta_{n-1} + \alpha_{s_n} + \frac{\sigma^2_\theta}{2}\left[y_n^2\exp(-\phi\theta_{n-1}-\alpha_{s_n})-1\right],$$
$$\sigma^2_{q_n} = \sigma^2_\theta.$$

Furthermore, we employ the following approximation of the predictive likelihood

$$\begin{aligned}\widehat{p}(y_n|\theta_{n-1},s_n) &\propto \int \widehat{g}(y_n|\theta_n;\theta_{n-1},s_n)r(\theta_n|\theta_{n-1},s_n)\mathrm{d}\theta_n \\ &\propto \exp\left(\frac{1}{2\sigma^2_\theta}(\mu^2_{q_n}-(\phi\theta_{n-1}+\alpha_{s_n})^2)\right) \\ &\quad\times\exp\left(-\frac{y_n^2}{2}\exp(-\phi\theta_{n-1}-\alpha_{s_n})(1+\phi\theta_{n-1}+\alpha_{s_n})\right).\end{aligned}$$

The importance weight is given by

$$\begin{aligned}\widetilde{w}_n(\theta_{n-1:n},s_n) &\propto \frac{g(y_n|\theta_n)}{\widehat{g}(y_n|\theta_n;\theta_{n-1},s_n)} \\ &\propto \exp\left\{-\frac{y_n^2}{2}\left[\exp(-\theta_n)-\exp(-\bar{\theta}_n)[1-(\theta_n-\bar{\theta}_n)]\right]\right\}.\end{aligned}$$

Due to the fact that $\log g(y_n|\theta)$ is concave as a function of θ and from the definition (3.29), the importance weight $\widetilde{w}_n(\theta_{n-1:n},s_n)$ is bounded above.

The Bovespa Index (IBOVESPA) is an index of approximately 50 stocks traded on the São Paulo Stock Exchange. Figure 3.3 shows weekday returns on the IBOVESPA index for the period 1/2/97–1/15/01. As highlighted in Carvalho and Lopes (2007), during this period there occurred several international currency events which affected Latin American markets, generating higher levels of uncertainty and consequently higher levels of volatility. These events are listed in table 3.4 and are indicated by the vertical dotted lines in figure 3.3. This data set was analysed in Lopes and Carvalho (2007), where an SMC algorithm was used to perform filtering whilst simultaneously estimating static parameters. We concentrate on the filtering problem and set static parameters to pre-determined values.

The sAPF was compared to a standard APF for this model, with the latter employing the same approximation of the likelihood in the proposal distributions, i.e.

$$q_n(\theta_n,s_n|\theta_{n-1},s_{n-1}) \propto \widehat{g}(y_n|\theta_n;\theta_{n-1},s_n)r(\theta_n|\theta_{n-1},s_n)r(s_n|s_{n-1}),$$
$$\widehat{p}(y_n|\theta_{n-1},s_{n-1}) \propto \sum_{j=1}^{M}\left\{r(s_n=j|s_{n-1})\right.$$
$$\left.\times\int \widehat{g}(y_n|\theta_n;\theta_{n-1},s_n=j)r(\theta_n|\theta_{n-1},s_n=j)d\theta_n\right\}.$$

Systematic resampling was used in both algorithms[3]. Based on the parameter estimates made in Carvalho and Lopes (2007), we set $M=2$, $p_{11}=0.993$, $p_{22}=0.973$, $\alpha_1=-1.2$, $\alpha_2=-0.9$, $\phi=0.85$, $\sigma^2_\theta=0.1$.

[3]Although systematic resampling does not uniformly outperform other approaches it is extremely widely used in the applied filtering literature. Although it is computationally attractive, care is required when using this approach for the reasons documented in Douc et al. (2005).

07/02/1997	Thailand devalues the Baht by as much as 20%
08/11/1997	IMF and Thailand set a rescue agreement
10/23/1997	Hong Kong's stock index falls 10.4%
	South Korean Won starts to weaken
12/02/1997	IMF and South Korea set a bailout agreement
06/01/1998	Russia's stock market crashes
06/20/1998	IMF gives final approval to a loan package to Russia
08/19/1998	Russia officially falls into default
10/09/1998	IMF and World Bank joint meeting to discuss the economic crisis
	The Federal Reserve cuts interest rates
01/15/1999	The Brazilian government allows its currency, the Real,
	to float freely by lifting exchange controls
02/02/1999	Arminio Fraga is named President of Brazil's Central Bank

Table 3.4: Events which impacted Latin American markets (Carvalho and Lopes, 2007)

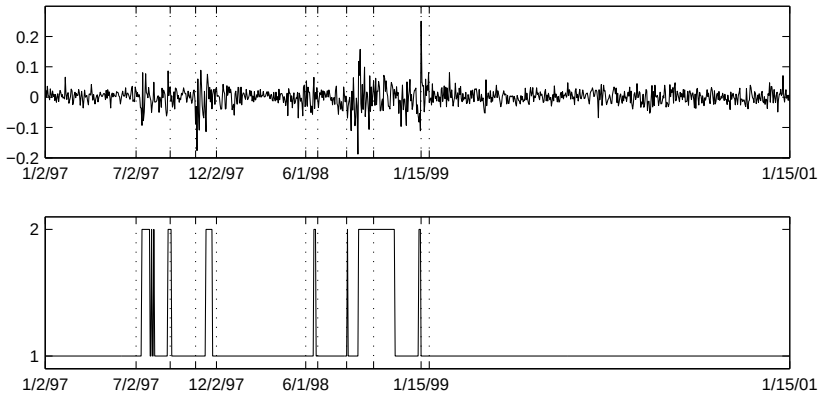


Figure 3.3: Stochastic Volatility Model. Top: Daily returns on the IBOVESPA index from February 1997 to January 2001. Bottom: MAP one–step–ahead prediction of the switching state s_n. State 2 is the high–volatility regime.

For each algorithm, the variance of the minimum means square error (MMSE) filtering estimate of the log volatility was computed at each iteration, over 500 independent runs. These variances were then summarised by taking their arithmetic mean and are shown in table 3.5.

	APF		**sAPF**	
$\boldsymbol{N}$	$\overline{\boldsymbol{\sigma^2}}$	**CPU / s**	$\overline{\boldsymbol{\sigma^2}}$	**CPU / s**
10	0.0906	0.8394	0.0850	0.2526
20	0.0544	1.5397	0.0492	0.3558
50	0.0325	3.6665	0.0290	0.6648
100	0.0274	10.7095	0.0230	1.1801
200	0.0195	17.7621	0.0189	2.7231
500	0.0195	35.4686	0.0185	5.3206

Table 3.5: Stochastic Volatility Model: variance of filtering estimate and average CPU time per run over 500 runs for the IBOVESPA data.

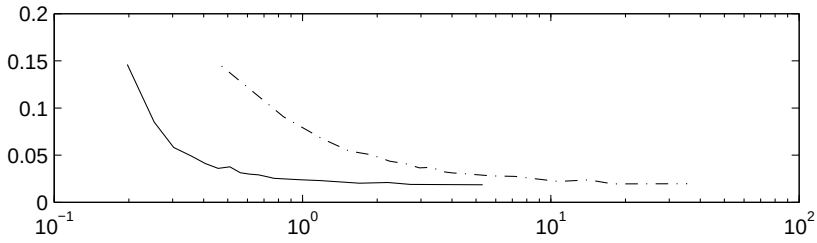


Figure 3.4: Stochastic Volatility Model: variance of filtering estimate vs average CPU time in secs. over 500 runs for the IBOVESPA data. Solid: sAPF, dash-dot: APF.

This shows how the variance of filtering estimates of the log–volatility and mean CPU time per run for the two algorithms relate to the number of particles used. For the same number of particles, the sAPF algorithm exhibits lower variance of filtering estimates. The results also show that, for the same number of particles, the sAPF can be computationally cheaper than the APF. This can be explained as follows. The algorithms involve precisely the same arithmetic operations in order to compute both the auxiliary importance weights and the importance weights by which estimation is performed. However, in terms of random number generation, the APF is more expensive: it uses one random variate to perform systematic resampling, then for each particle draws $S_n^{(i)}$ from a distribution on $\{1, ..., M\}$ and samples $\theta_n^{(i)}$ from $q_n(\cdot|\theta_{n-1}, s_n)$. By contrast, the sAPF uses one random variate to perform systematic resampling (which assigns values of both $X_{n-1}^{(i)}$ *and* $S_n^{(i)}$) and then for each particle samples $\theta_n^{(i)}$ from $q_n(\cdot|\theta_{n-1}, s_n)$.

Although the cost–saving will be dependent on the programming language employed, the results indicate that the savings can be significant. In this case both algorithms were implemented in MatLab, and the code was made common to both algorithms in all places possible. The performance benefit in terms of estimator variance versus CPU time is illustrated in figure 3.4.

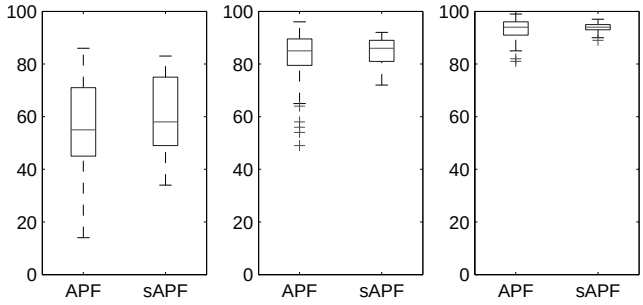


Figure 3.5: Stochastic Volatility Model. Boxplots, over 100 runs of each algorithm, of the number of particles in the high–volatility regime at iterations corresponding to the dates 1/13/99 (left), 1/14/99 (middle) and 1/15/99 (right). $N = 100$.

Figure 3.5 shows boxplots of the number of particles in the high–volatility regime over 100 independent runs of each algorithm. The pairs of boxplots correspond to the dates 1/13/99 (left), 1/14/99 (middle) and 1/15/99 (right). During this period, it can be seen from figure 3.3 that an increase in volatility occurs. $N = 100$ particles were used in both algorithms. The count of number of particles in the high–volatility regime was made immediately after resampling in the case of the sAPF and immediately after making proposals in the case of the APF, i.e. at equivalent

steps of the algorithms. Across the three dates the sAPF exhibits lower variability than the APF and the mean number of particles in the high–volatility regime is lower for the APF. That is, the sAPF shows less variability in its approximation of the distribution over strata: this improved distributional approximation is the underlying mechanism which leads to improved variance properties.

Figure 3.3 shows the one-step-ahead MAP prediction of the switching state s_n, using the sAPF algorithm with $N = 500$ particles. Recall that $s_n = 2$ is the high volatility regime. The results show that the model is able to recognise changes in the level of volatility and these changes roughly coincide with the currency crisis events listed in table 3.4. The results are very similar to those obtained in Carvalho and Lopes (2007).

3.5 Conclusions

This article has attempted to summarise the state of the art of the auxiliary particle filter. Our intention is to provide some insight into the behaviour of the APF and its relationship with other particle-filtering algorithms, in addition to summarising a number of recent methodological extensions. One of the most significant points is perhaps this: the APF is simply an example of a sequential estimation procedure in which one can benefit from the early introduction of information about subsequent distributions, combined with an importance sampling correction. In the context of time series analysis, this approach is useful when performing filtering in SSMs and the same approach can be exploited elsewhere.

Bibliography

Andrieu, C., M. Davy, and A. Doucet (2003). Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions. *IEEE Transactions on Signal Processing 51*(7), 1762–1770.

Bain, A. and D. Crisan (2009). *Fundamentals of Stochastic Filtering*. Stochastic Modelling and Applied Probability. Springer Verlag.

Briers, M., A. Doucet, and S. S. Singh (2005). Sequential auxiliary particle belief propagation. In *Proceedings of International Conference on Information Fusion*.

Cappé, O., E. Moulines, and T. Ryden (2005). *Inference in Hidden Markov Models*. New York: Springer Verlag.

Carpenter, J., P. Clifford, and P. Fearnhead (1999). An improved particle filter for non-linear problems. *IEEE Proceedings on Radar, Sonar and Navigation 146*(1), 2–7.

Carvalho, C. M. and H. F. Lopes (2007). Simulation-based sequential analysis of Markov switching stochastic volatility models. *Computational Statistics and Data Analysis 51*, 4526–4542.

Centanni, S. and M. Minozzo (2006). Estimation and filtering by reversible jump MCMC for a doubly stochastic Poisson model for ultra-high-frequency financial data. *Statistical Modelling 6*(2), 97–118.

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its applications to Bayesian inference. *Annals of Statistics 32*(6), 2385–2411.

Clark, D. E. and J. Bell (2006, July). Convergence results for the particle PHD filter. *IEEE Transactions on Signal Processing 54*(7), 2652–2661.

Cornebise, J., E. Moulines, and J. Olsson (2008). Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing 18*, 461–480.

Daley, D. J. and D. Vere-Jones (2003). *An Introduction to the Theory of Point Processes* (Second ed.), Volume I: Elementary Theory and Methods of *Probability and Its Applications*. New York: Springer.

Dassios, A. and J. Jang (2005). Kalman-Bucy filtering for linear system driven by the Cox process with shot noise intensity and its application to the pricing of reinsurance contracts. *Journal of Applied Probability 42*(1), 93–107.

Del Moral, P. (2004). *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Probability and Its Applications. New York: Springer Verlag.

Del Moral, P., A. Doucet, and A. Jasra (2006a). Sequential Monte Carlo methods for Bayesian Computation. In *Bayesian Statistics 8*. Oxford University Press.

Del Moral, P., A. Doucet, and A. Jasra (2006b). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society Series B 63*(3), 411–436.

Del Moral, P., A. Doucet, and A. Jasra (2008). On adaptive resampling procedures for sequential monte carlo methods. Technical Report HAL-INRIA RR-6700, INRIA.

Douc, R., O. Cappé, and E. Moulines (2005). Comparison of resampling schemes for particle filters. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, Volume I, pp. 64–69.

Douc, R. and E. Moulines (2008). Limit theorems for weighted samples with applications to sequential Monte Carlo methods. *Annals of Statistics 36*(5), 2344–2376.

Douc, R., E. Moulines, and J. Olsson (2009). Optimality of the auxiliary particle filter. *Probability and Mathematical Statistics 29*(1), 1–28.

Doucet, A., N. de Freitas, and N. Gordon (Eds.) (2001). *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. New York: Springer Verlag.

Doucet, A., N. Gordon, and V. Krishnamurthy (2001). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing 49*(3), 613–624.

Doucet, A. and A. M. Johansen (2009). A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky (Eds.), *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press. To appear.

Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory*. Ph. D. thesis, University of Oxford.

Fearnhead, P., O. Papaspiliopoulos, and G. O. Roberts (2008). Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society B 70*, 755–777.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica 57*(6), 1317–1339.

Gilks, W. R. and C. Berzuini (2001). RESAMPLE-MOVE filtering with Cross-Model jumps. See Doucet et al. (2001), pp. 117–138.

Godsill, S. and T. Clapp (2001). Improvement strategies for Monte Carlo particle filters. See Doucet et al. (2001), pp. 139–158.

Godsill, S. J., J. Vermaak, K.-F. Ng, and J.-F. Li (2007, April). Models and algorithms for tracking of manoeuvring objects using variable rate particle filters. *Proc. IEEE*.

Gray, A. G. and A. W. Moore (2000). N-body problems in statistical learning. In *Advances in Neural Information Processing Systems 13*, pp. 521–527. MIT Press.

Heine, K. (2005). Unified framework for sampling/importance resampling algorithms. In *Proceedings of Fusion 2005, July 25-29, 2005, Philadelphia*.

Johansen, A. M. (2009). SMCTC: Sequential Monte Carlo in C++. *Journal of Statistical Software 30*(6), 1–41.

Johansen, A. M. and A. Doucet (2007). Auxiliary variable sequential Monte Carlo methods. Research Report 07:09, University of Bristol, Department of Mathematics – Statistics Group, University Walk, Bristol, BS8 1TW, UK.

Johansen, A. M. and A. Doucet (2008). A note on the auxiliary particle filter. *Statistics and Probability Letters 78*(12), 1498–1504.

Johansen, A. M., S. Singh, A. Doucet, and B. Vo (2006). Convergence of the SMC implementation of the PHD filter. *Methodology and Computing in Applied Probability 8*(2), 265–291.

Kantas, N. (2009). *Sequential Decision Making in General State Space Models*. Ph. D. thesis, University of Cambridge.

Karlsson, R. and N. Bergman (2000). Auxiliary particle filters for tracking a maneuvering target. In *Proceedings of the 39th IEEE Conference on Decision and Control*, pp. 3891–3895.

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

Klass, M., N. de Freitas, and A. Doucet (2005). Towards practical n^2 Monte Carlo: The marginal particle filter. In *Proceedings of Uncertainty in Artificial Intelligence*.

Liu, J. S. and R. Chen (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association 93*(443), 1032–1044.

Lopes, H. F. and C. M. Carvalho (2007). Factor stochastic volatility with time varying loadings and Markov switching regimes. *Journal of Statistical Planning and Inference 137*, 3082–3091.

Mahler, R. P. S. (2003, October). Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 1152–1178.

Mahler, R. P. S. (2007). *Statistical Multisource-Multitarget Information Fusion*. Artech House.

Nadarajah, S. and S. Kotz (2005). Mathematical properties of the multivariate t distribution. *Acta Applicandae Mathematicae 89*, 53–84.

Pitt, M. K. and N. Shephard (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association 94*(446), 590–599.

Pitt, M. K. and N. Shephard (2001). Auxiliary variable based particle filters. See Doucet et al. (2001), Chapter 13, pp. 273–293.

Poyiadjis, G., A. Doucet, and S. Singh (2005). Particle methods for optimal filter derivative: application to parameter estimation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 5, pp. 925–928.

Robert, C. P. and G. Casella (2004). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer.

Saha, S., P. K. Mandal, Y. Boers, H. Driessen, and A. Bagchi (2009). Gaussian proposal density using moment matching in SMC methods. *Statistics and Computing 19*, 203–208.

Sidenbladh, H. (2003). Multi-target particle filtering for the probability hypothesis density. In *Proceedings of the International Conference on Information Fusion, Cairns, Australia*, pp. 800–806.

Singh, S., B.-N. Vo, A. Baddeley, and S. Zuyev (2008). Filters for spatial point processes. *Siam Journal on Control and Optimization 48*(4), 2275–2295.

So, M. K. P., K. Lam, and W. K. Li (1998). A stochastic volatility model with Markov switching. *Journal of Business and Economic Statistics 16*(2), 244–253.

Vo, B., S. Singh, and A. Doucet (2005). Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems 41*(4), 1224–1245.

Whiteley, N., A. M. Johansen, and S. Godsill (2009). Monte Carlo filtering of piecewise-deterministic processes. *Journal of Computational and Graphical Statistics*. To appear.

Whiteley, N., S. Singh, and S. Godsill (2009). Auxiliary particle implementation of the probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*. To Appear.

Zajic, T. and R. P. S. Mahler (2003). Particle-systems implementation of the PHD multitarget tracking filter. In *Proceedings of SPIE*, pp. 291–299.