Bernoulli **19**(4), 2013, 1391–1403
DOI: 10.3150/12-BEJSP07

Particle filters

HANS R. KÜNSCH

Seminar für Statistik, D-Math, ETH Zurich, CH-8092 Zurich, Switzerland.
E-mail: kuensch@stat.math.ethz.ch

This is a short review of Monte Carlo methods for approximating filter distributions in state space models. The basic algorithm and different strategies to reduce imbalance of the weights are discussed. Finally, methods for more difficult problems like smoothing and parameter estimation and applications outside the state space model context are presented.

Keywords: Ensemble Kalman filter; importance sampling and resampling; sequential Monte Carlo; smoothing algorithm; state space models

1. Introduction

Filtering is engineering terminology for extracting information about a signal from partial and noisy observations. In geophysics, filtering is usually called data assimilation. In the last 50 years, filtering has been mainly studied in the framework of state space or hidden Markov models, assuming a Markovian time evolution of the signal and observations which are instantaneous functions of the signal subject to white observation noise. Developments started in the 1960s with the Kalman–Bucy filter (Kalman (1960), Kalman and Bucy (1961)) for linear Gaussian models and with the forward–backward algorithm due to Baum and Welch for models with a finite state space (see p. 74 of Cappé, Moulines and Rydén (2005) for the history of this algorithm, including references). The essential feature of these methods is that they are recursive and thus suitable for online applications where the observations arrive sequentially and quantities of interest have to be recomputed with each new observation.

Probabilists started in the mid-sixties to develop a general theory of nonlinear filtering in continuous time. In statistics, state space models and filtering techniques took longer to take roots. In the seventies and eighties, the relation between linear state space and ARMA models was studied and used. A breakthrough occurred with the paper Gordon, Salmond and Smith (1993) which developed recursive Monte Carlo methods called particle filters. Interestingly, Handschin and Mayne (1969) had proposed much earlier to use Monte Carlo methods, but the idea of resampling was missing. However, this idea is essential to ensure that the required sample size for a given accuracy does not explode with the number of time steps. Particle filters quickly became very popular. Among other things they have also been used for continuous time filtering. Nowadays, they are also applied outside the context of state space models as a complement to other, static MCMC methods. In the 1990s, geophysicists developed a different version of the particle filter, called the Ensemble Kalman filter which is more stable in high dimensions. After some delay, this idea has now also become part of the research in statistics.

There are many presentations of the topic in books and in survey articles (e.g., Künsch (2001), Doucet, de Freitas and Gordon (2001), Del Moral (2004), Cappé, Moulines and Rydén

1350-7265

(2005), Cappé, Godsill and Moulines (2007), Crisan and Rozovskiĭ (2011), Doucet and Johansen (2011)). This paper gives a brief introduction for non-specialists, explaining the main algorithms, describing their scope and also their limitations and surveying some of the interesting current developments. Because of limitations of space, many interesting topics and references that would deserve to be mentioned had to be omitted.

2. State space models

2.1. Definitions

A state space model consists of an unobservable $(S, \mathcal{S})$-valued Markov process (X_t), the state of a system or the signal, combined with partial and noisy $\mathbb{R}^d$-valued observations $(Y_i; i \geq 1)$ of the state at discrete times t_i. In order to simplify the notation, we assume $t_i = i$. We denote the initial distribution of X_0 by π_0 and the conditional distribution of X_i given $X_{i-1} = x_{i-1}$ by $P(dx_i|x_{i-1})$. Observations at different times are assumed to be conditionally independent given the states, and the conditional distributions of Y_i given X_i are assumed to have densities g with respect to some reference measure ν (usually the Lebesgue or the counting measure). Time homogeneity of these conditional distributions is only assumed to simplify notation.

The state process can be in continuous or discrete time. In the former case, the transition kernel P is usually not available analytically. For some of the algorithms, this is not necessary, it is sufficient that we are able to simulate from $P(\cdot|x)$ for any value x. Because some applications have a deterministic or partially deterministic state evolution, we do not assume the existence of densities for P.

Throughout, notation like $X_{0:n}$ for $(X_0, \ldots, X_n)$ is used. By a slight abuse of notation, p stands for any (conditional) density: The arguments of p will indicate which random variables are involved. The ratio of two probability measures is an abbreviation for the Radon–Nikodym derivative.

2.2. Examples

State space models have a wide range of applications in finance (stochastic volatility, interest rates), engineering (tracking, speech recognition, computer vision), biology (genome sequence analysis, ion channels, stochastic kinetic models), geophysics (meteorology, oceanography, reservoir modeling), analysis of longitudinal data and others. It is not possible here to describe these applications in detail or give references to all relevant pulications. Some of these applications are discussed in Künsch (2001) and in Doucet, de Freitas and Gordon (2001). A few references of more recent applications are Bretó *et al.* (2009) and Wilkinson (2011) for biology, Part IX in Crisan and Rozovskiĭ (2011) for financial mathematics, and Evensen (2007), Aanonsen *et al.* (2009) and Cressie and Wikle (2011) for geophysical applications.

3. The basic particle filter

3.1. Filtering recursions

By the assumptions on the state and observation process, we have the following joint distributions for $n \geq m$

$$(X_{0:n}, Y_{1:m}) \sim \pi_0(dx_0) \prod_{t=1}^{n} P(dx_t|x_{t-1}) \prod_{t=1}^{m} g(y_t|x_t)\nu(dy_t). \tag{1}$$

The information about the state contained in the observations is expressed by the conditional distributions $\pi_{s:t|n}$ of $X_{s:t}$ given $Y_{1:n} = y_{1:n}$. Of particular interest are $\pi_{0:n} := \pi_{0:n|n}$, called here the joint smoothing distribution, and $\pi_n := \pi_{n|n}$, called here the filter distribution (the terminology is not unique). For $n \geq m$, $\pi_{0:n|m}$ follows immediately from (1) and Bayes formula. Other cases are then obtained in principle by marginalization. We are however interested in methods to compute or approximate expectations with respect to these distributions in an explicit and efficient way. For this, recursive formulae are most useful. It is straightforward to verify that

$$\pi_{0:n|n-1}(dx_{0:n}|y_{1:n-1}) = \pi_{0:n-1}(dx_{0:n-1}|y_{1:n-1})P(dx_n|x_{n-1}), \tag{2}$$

$$\pi_{0:n}(dx_{0:n}|y_{1:n}) = \pi_{0:n|n-1}(dx_{0:n}|y_{1:n-1}) \frac{g(y_n|x_n)}{p_n(y_n|y_{1:n-1})}, \tag{3}$$

where

$$p_n(y_n|y_{1:n-1}) = \int \pi_{n|n-1}(dx_n|y_{1:n-1})g(y_n|x_n). \tag{4}$$

By marginalization, we therefore also have the recursions

$$\pi_{n|n-1}(dx_n|y_{1:n-1}) = \int \pi_{n-1}(dx_{n-1}|y_{1:n-1})P(dx_n|x_{n-1}), \tag{5}$$

$$\pi_n(dx_n|y_{1:n}) = \pi_{n|n-1}(dx_n|y_{1:n-1}) \frac{g(y_n|x_n)}{p_n(y_n|y_{1:n-1})}. \tag{6}$$

In both cases, the recursions consist of a propagation step (2) or (5), respectively, and an update or correction step, (3) or (6), respectively. Typically, one wants to compute these recursions for an arbitrary, but fixed sequence $y_1, y_2, \ldots$ (not necessarily a realization from the state space model).

3.2. Analytical solutions

There are two important special cases where one can perform the above recursions exactly. In the first one, the state space S is finite and the integrals reduce to finite sums which can be computed with $O(n|S|^2)$ operations. The second special case are linear Gaussian state space models where $X_n|X_{n-1} \sim \mathcal{N}(FX_{n-1}, V)$ and $Y_n|X_n \sim \mathcal{N}(HX_n, R)$. If π_0 is also Gaussian, then all π_n are Gaussian and (5)–(6) lead to recursions for the conditional means and covariances. For

comparison with the Ensemble Kalman filter below, we write down the update step for going from $\pi_{n|n-1} = \mathcal{N}(m_{n|n-1}, P_{n|n-1})$ to $\pi_n = \mathcal{N}(m_n, P_n)$:

$$m_n = m_{n|n-1} + K_n(Y_n - Hm_{n|n-1}), \quad P_n = P_{n|n-1} - K_n H P_{n|n-1} \tag{7}$$

where $K_n = P_{n|n-1}H'(HP_{n|n-1}H' + R)^{-1}$ is the so-called Kalman gain.

In most other cases of practical interest, one has to approximate the integrals involved in (5) and (4). Numerical approximations are difficult to use because the region of main mass of π_n changes with n and is unknown in advance. The particle filter tries to generate values in this region adaptively as new observations arise.

3.3. The particle filter

The particle filter recursively computes importance sampling approximations of π_n, that is

$$\pi_n(dx_n|y_{1:n}) \approx \hat{\pi}_n(dx_n|y_{1:n}) = \sum_{i=1}^{N} W_n^i \Delta_{X_n^i}(dx_n).$$

Here the W_n^i are random weights which sum to one, X_n^i are random variables called "particles" and Δ_x is the point mass at x. At time 0, we draw particles from π_0 and set $W_0^i = 1/N$. At time n we start with $\hat{\pi}_{n-1}$ and draw independently new particles X_n^i from $P(\cdot|X_{n-1}^i)$. By (5), the particles X_n^i with weights W_{n-1}^i provide an importance sampling approximation of $\pi_{n|n-1}$. If we also update the weights with $W_n^i \propto W_{n-1}^i g(y_n|X_n^i)$, we have closed the recursion by (6).

This algorithm has the drawback that after a few iterations most particles are located at positions very far away from the region of main mass of π_n and the weights are very unbalanced. This can be avoided by introducing a resampling step before propagation such that particles with low weights die and particles with high weights have much offspring that is independently propagated afterwards. Thus the basic particle filter, also called the bootstrap filter or SIR-filter (Sampling Importance Resampling), works as follows.

Algorithm 1. 1. *Resample*: *Draw* $(X_{n-1}^{*1}, \ldots, X_{n-1}^{*N})$ *from* $\hat{\pi}_{n-1}$.
2. *Propagate*: *Draw* X_n^i *from* $P(\cdot|X_{n-1}^{*i})$, *independently for different indices* i.
3. *Reweight*: *Set* $W_n^i \propto g(y_n|X_n^i)$.

Note that for any function $\varphi : S \to \mathbb{R}$, $N^{-1}\sum_i \varphi(X_{n-1}^{*i})$ always has a larger variance than $\sum_i W_{n-1}^i \varphi(X_{n-1}^i)$. The advantage of resampling is seen only after one or several propagation steps. Because of this, we resample at the beginning and not at the end of a recursion.

As a byproduct, the particle filter gives also the following estimate of (4)

$$\hat{p}_n(y_n|y_{1:n-1}) = \frac{1}{N}\sum_{i=1}^{N} g\big(y_n|X_n^i\big).$$

One can show by induction that the product $\prod_{t=1}^{n} \hat{p}_t(y_t|y_{1:t-1})$ is an exactly unbiased estimator of $p(y_{1:n}) = \prod_{t=1}^{n} p_t(y_t|y_{1:t-1})$ for any n and any $y_{1:n}$, see Theorem 7.4.2 in Del Moral (2004). However, $\hat{p}_t$ is in general not unbiased for p_t.

3.4. Simple improvements

Because the numbering of particles is irrelevant, we only need to know the number of times N_n^i that the i-th particle is selected in the resampling step. One can therefore reduce the additional variability introduced by resampling by a so-called balanced resampling scheme, meaning that $\mathbb{E}(N_n^i) = NW_n^i$ and $|N_n^i - NW_n^i| < 1$. The simplest such scheme uses a uniform$(0, 1)$ random variable U and takes as N_n^i the number of points in the intersection of $(U + \mathbb{Z})/N$ with $(\sum_{k=1}^{i-1} W_n^k, \sum_{k=1}^{i} W_n^k]$. See Crisan (2001) for other balanced resampling schemes. Since balanced resampling can always be used at little extra cost, it is widely used.

A second improvement omits the resampling step whenever the weights are sufficiently uniform. As criterion, one often uses the so-called effective sample size which is defined as one over $\sum_{i=1}^{N} (W_n^i)^2$, see Liu (1996) for a justification of the name of this criterion.

In the propagation step, we can draw X_n^i not from $P(\cdot|X_{n-1}^{*i})$, but from any other distribution Q which dominates $P(\cdot|X_{n-1}^{*i})$. We then have to adjust the weights in the reweighting step. The correct weights are obtained by setting $r \equiv 1$ in step 4 of Algorithm 2 below. By letting Q depend not only on X_{n-1}^{*i}, but also on the new observation y_n, we can make the propagated particles X_n^i more compatible with y_n and thus the weights more balanced. In the so-called auxiliary particle filter due to Pitt and Shephard (1999), one uses the new observation y_n not only in the propagation step, but also in an additional reweighting step before resampling. The goal of this additional reweighting is to bring $\hat{\pi}_{n-1}$ closer to $\pi_{n-1|n}$. Thus, the auxiliary particle filter works as follows.

Algorithm 2. 1. *Reweight*: *Set*

$$\hat{\pi}_{n-1|n} = \sum_{i=1}^{N} W_{n-1}^{*i} \Delta_{X_{n-1}^i}(dx_{n-1})$$

where $W_{n-1}^{*i} \propto W_{n-1}^i r(X_{n-1}^i, y_n)$.

2. *Resample*: *Draw* $(X_{n-1}^{*1}, \ldots, X_{n-1}^{*N})$ *from* $\hat{\pi}_{n-1|n}$.

3. *Propagate*: *Draw* X_n^i *from* $Q(\cdot|X_{n-1}^{*i}, y_n)$, *independently for different indices* i.

4. *Reweight*: *Set*

$$W_n^i \propto w_n^i := \frac{g(y_n|X_n^i)}{r(X_{n-1}^{*i}, y_n)} \frac{P(dx_n|X_{n-1}^{*i})}{Q(dx_n|X_{n-1}^{*i}, y_n)}(X_n^i).$$

In order to understand the formula for w_n^i, note that (X_{n-1}^{*i}, X_n^i) has distribution proportional to $\hat{\pi}_{n-1}(dx_{n-1}) r(x_{n-1}, y_n) Q(dx_n|x_{n-1}, y_n)$ and the distribution target is proportional to

$\hat{\pi}_{n-1}(dx_{n-1})g(y_n|x_n)P(dx_n|x_{n-1})$. Because the average of the unnormalized weights w_n^i estimates the ratio of the normalizing constants, the estimate of (4) is now

$$\hat{p}_n(y_n|y_{1:n-1}) = \frac{1}{N}\sum_{i=1}^{N} w_n^i \cdot \sum_{k=1}^{N} r\big(X_{n-1}^k, y_n\big) W_{n-1}^k.$$

The product $\prod_{t=1}^{n} \hat{p}_t(y_t|y_{1:t})$ is again unbiased for $p(y_{1:n})$.

Auxiliary particle filters cannot be used if the state evolution is deterministic, or if the density $dP(\cdot|x')/dQ(\cdot|x', y)$ is not available in closed form. In other cases, the choices of r and Q are up to the user. Ideally, we take $r(x, y) = \int g(y|x')P(dx'|x)$ and $Q(dx'|x, y) = r(x, y)^{-1}g(y|x')P(dx'|x)$, because then the weights W_n^i in the fourth step are constant. In most cases, these choices are not possible, but one can try to find suitable approximations. With the ideal choices for r and Q, the auxiliary particle filter therefore leads to a reweighting with $p(y_n|x_{n-1})$ instead of $p(y_n|x_n)$: Although this usually reduces the variance of the weights, the gain may not be substantial. In principle, it is possible to go further back in time by computing particle filter approximations of $\pi_{n-L:n|n}$ for some $L > 0$. An auxiliary particle filter in this case uses $y_{n-L:n}$ to reweight the particles at time $n - L - 1$ and to generate new particles at times $n - L$ to n.

4. Complications and solutions

4.1. Main difficulties

The main difficulty with the particle filter is that often weights become unbalanced, even when we use the auxiliary particle filter in Algorithm 2 or apply some of the other simple improvements discussed above. In such cases, most resampled particles coincide ("sample depletion"). If the state transitions are partially deterministic, this becomes especially drastic because the propagation will not create diversity.

Partially deterministic state transitions occur for instance if the model contains unknown parameters θ in the state transition P or in the observation density g and one proceeds by considering the enlarged state vector (θ, X_t). The propagation step for θ is then simply $\theta_n^i = \theta_{n-1}^{*i}$. One can add some noise to create diversity, possibly combined with some shrinking towards the mean to keep the variance the same. Still, this does not always work well.

A second instance with partially deterministic state transitions occurs if one uses the particle filter algorithm to approximate not only π_n, but the whole smoothing distribution $\pi_{0:n}$. In principle, this is straightforward: Each particle at time n is then a path of length $n + 1$ that we write as $X_{0:n|n}^i$. The propagation step concatenates a resampled path $X_{0:n-1|n-1}^{*i}$ with a new value $X_n^i \sim P(\cdot|X_{n-1|n-1}^{*i})$.

If the weights at one time point become very unbalanced, the filter can be completely unreliable and it can lose track even though the propagation step later creates again diversity. Unbalanced weights have been observed to occur easily if the dimension of the observations is large. A theoretical explanation of this phenomenon has been provided by Bickel, Li and Bengtsson (2008).

In the following, we discuss some more advanced methods that have been proposed to overcome these difficulties.

4.2. Resample moves

Gilks and Berzuini (2001) have proposed the following method to avoid sample depletion when the particle filter is used to produce an approximation of $\pi_{0:n}$ with particles $X^i_{0:n|n}$ and equal weights. Let K_n be a Markov kernel on S^{n+1} which has $\pi_{0:n}$ as invariant distribution, constructed for instance according to the general Metropolis–Hastings recipe. Drawing new particles $X^{*i}_{0:n|n} \sim K_n(.|X^i_{0:n|n})$, independently for different i's will then give a new approximation of $\pi_{0:n}$ which is expected to be at least as good as the old one. If K_n modifies all components of $X^i_{0:n|n}$, this method also removes ties, but since typically a single kernel can only update one or a few components of $X^i_{0:n|n}$, the computational complexity increases with n if one wants to get rid of all ties.

4.3. Ensemble Kalman filter

This method is due to Evensen (1994). It assumes linear observations with Gaussian errors, that is, $g(\cdot|x)$ is a normal density with mean Hx and variance R. It uses particles with equal weights, the propagation step is the same as in the particle filter whereas the update step is a Monte Carlo implementation of the Kalman filter update (7) with estimated first and second moment of $\pi_{n|n-1}$:

Algorithm 3. 1. *Propagate*: *Draw* X^{*i}_n *from* $P(\cdot|X^i_{n-1})$.

2. *Update*: *Draw i.i.d. values* $\varepsilon^i_n \sim \mathcal{N}(0, R)$ *and set*

$$X^i_n = X^{*i}_n + \widehat{K}_n\big(y_n - HX^{*i}_n + \varepsilon^i_n\big)$$

where $\widehat{K}_n$ *is the Kalman gain computed with the sample covariance* $\widehat{P}_{n|n-1}$ *of the* X^{*i}_n*'s.*

It is not difficult to show that the algorithm is consistent as $N \to \infty$ for a linear Gaussian state space model. However, for non-Gaussian $\pi_{n|n-1}$, this update typically has a systematic error because only the location, but neither the spread nor the shape of the sample (X^i_n) change if y_n changes. Nevertheless, the Ensemble Kalman filter is extremely wide-spread in geophysical applications where the state evolution is usually complicated, making the propagation step the computational bottleneck. This forces one to use a sample size N which is much smaller than the dimensions of the state or the observation. Even in such cases, the Ensemble Kalman filter turns out to be surprisingly robust – provided we regularize the estimate $\widehat{P}_{n|n-1}$ of the covariance of $\pi_{n|n-1}$.

Several attempts have been made to find algorithms which combine the robustness of the Ensemble Kalman filter with the nonparametric features of the particle filter. They either approximate $\pi_{n|n-1}$ by a mixture of Gaussians or use the Ensemble Kalman filter as a proposal

distribution Q in a particle filter. See Frei and Künsch (2012) for references and a new proposal which avoids both the fitting of a Gaussian mixture to the forecast sample (X_n^{*i}) and the estimation of the density dP/dQ (which is usually not known analytically in these applications).

An extension of the Ensemble Kalman filter to more general observation densities g has been given in Lei and Bickel (2011).

4.4. Particle smoothing

In an offline application where all T observations are available from the beginning, one can use smoothing algorithms which combine a forward filtering pass through the data from $n = 0$ to $n = T$ with a backward recursion from $n = T - 1$ to $n = 0$. We limit ourselves to approximations of the marginals $\pi_{n|T}$, but the same methods apply also for joint distributions.

By Bayes formula and conditional independence, we obtain the following relations

$$\pi_{n|T}(dx_n|y_{1:T}) = \pi_{n|n-1}(dx_n|y_{1:n-1})\frac{p(y_{n:T}|x_n)}{p(y_{n:T}|y_{1:n-1})} \tag{8}$$

$$= \pi_n(dx_n|y_{1:n})\frac{p(y_{n+1:T}|x_n)}{p(y_{n+1:T}|y_{1:n})}. \tag{9}$$

This is also called the two-filter formula because we have the recursions

$$p(y_{n+1:T}|x_n) = \int p(y_{n+1:T}|x_{n+1})P(dx_{n+1}|x_n), \tag{10}$$

$$p(y_{n:T}|x_n) = g(y_n|x_n)p(y_{n+1:T}|x_n) \tag{11}$$

which are dual to (5) and (6). Combining (8)–(9) with (10) gives

$$\frac{\pi_{n|T}(dx_n|y_{1:T})}{\pi_n(dx_n|y_{1:n})} \propto \int \frac{\pi_{n+1|T}(dx_{n+1}|y_{1:T})}{\pi_{n+1|n}(dx_{n+1}|y_{1:n})}P(dx_{n+1}|x_n). \tag{12}$$

In order to be able to use Monte Carlo methods, we have to assume that for any x' the state transition kernel $P(\cdot|x')$ has density $p(\cdot|x')$ with respect to some measure μ on S. Then the filter distributions also have densities which we denote by the same symbol. The right-hand side of (12) can then be considered as an integral with respect to $\pi_{n+1|T}$. Thus we obtain a marginal particle smoother $\hat{\pi}_{n|T}$ which has the same particles as the filter, but different weights $W_{n|T}^i$ which are computed with the recursion

$$W_{n|T}^i = \sum_{k=1}^N W_{n+1|T}^k \frac{W_n^i p(X_{n+1}^k|X_n^i)}{\sum_j W_n^j p(X_{n+1}^k|X_n^j)}.$$

The disadvantage is the complexity of the algorithm which is of the order $O(N^2)$.

The algorithm in Briers, Doucet and Maskell (2010) computes first backward particle approximations of the distributions $\bar{\pi}_n(dx_n|y_{n:T}) \propto p(y_{n:T}|x_n)h_n(x_n)\mu(dx_n)$ where h_n is a known

function such that $p(y_{n:T}|x_n)h_n(x_n)$ is integrable. Inserting a forward particle filter approximation for $\pi_{n|n-1}$ and a backward particle filter approximation for $p(y_{n:T}|x_n)$ into (8) gives then an approximation of $\pi_{n|T}$ which is concentrated on the particles approximating $\bar{\pi}_n$.

Fearnhead, Wyncoll and Tawn (2010) have suggested to insert particle approximations into

$$\pi_{n|T}(x_n|y_{1:T}) \propto \pi_{n|n-1}(x_n|y_{1:n-1})g(x_n|y_n) \int \frac{p(x_{n+1}|x_n)}{h_{n+1}(x_{n+1})} \bar{\pi}_{n+1}(dx_{n+1}|y_{n+1:T})$$

which follows by combining (8) with (10) and (11). This has the advantage that the support of $\hat{\pi}_{n|T}$ is not constrained on the sampled particles from the forward or the backward recursion. Moreover, one can sample from the approximation with an algorithm of complexity $O(N)$ which may not be efficient, however.

4.5. Particle MCMC

This is a recent innovation by Andrieu, Doucet and Holenstein (2010) which uses particle filters as a building block in an MCMC algorithm. Assume that g and the density of P both depend on an unknown parameter θ with prior density $p(\theta)$ and that we want to sample from the posterior $p(x_{0:T}, \theta|y_{1:T})$. A Gibbs sampler which updates single components of $x_{0:T}$ given the rest is usually too slow, and exact updates of the whole sequence $x_{0:T}$ are usually not possible. What the particle filter provides are *random approximations* $\hat{\pi}_{0:T;\theta}$ of $\pi_{0:T;\theta} = p(x_{0:T}|y_{1:T}, \theta)$ for any fixed θ. Andrieu, Doucet and Holenstein (2010) show that with these random approximations one can still construct Markov chains which leave the correct posterior invariant without letting the number of particles go to infinity.

The first such algorithm is called particle marginal Metropolis–Hastings sampler. It is an approximation of the sampler which jointly proposes $(\theta', x'_{0:T})$ from the distribution $q(\theta'|\theta)\, d\theta' \pi_{0:T}(dx'_{0:T}|y_{1:T}, \theta')$ with the acceptance ratio

$$\frac{p(y_{1:T}|\theta')p(\theta')q(\theta'|\theta)}{p(y_{1:T}|\theta)p(\theta)q(\theta|\theta')}.$$

The approximation occurs at two places: First $x'_{0:T}$ is generated from $\hat{\pi}_{0:T;\theta'}$ instead of $\pi_{0:T;\theta}$, and second the unknown likelihoods $p(y_{1:T}|\theta')$ and $p(y_{1:T}|\theta)$ in the acceptance ratio are replaced by unbiased estimates from the particle filter. The surprising result is that the errors from these two approximations cancel and the algorithm has the exact posterior $p(x_{0:T}, \theta|y_{1:T})$ as invariant distribution for any N.

Instead of jointly proposing a parameter and a path of the state process, one can also use a Gibbs sampler, alternating between updates of the parameter and the state process. Updating the parameter given the state and the observations is usually feasible, but for the other update one samples again from a particle filter approximation $\hat{\pi}_{0:T;\theta}$ and not from $\pi_{0:T;\theta}$. Andrieu, Doucet and Holenstein (2010) show that this also gives a correct algorithm for any $N > 1$ provided the particle filter approximation is modified such that the current path is equal to one of the particle paths $X^k_{0:T}$ in $\hat{\pi}_{0:T;\theta}$.

5. Convergence results

Laws of large numbers as well as central limit theorems have been shown for particle filter approximations. Del Moral (2004) contains general results, Künsch (2005) gives an essentially self-contained short derivation. First, one can show that for every n, every $y_{1:n}$ and a suitable class of functions φ, $\int \varphi(x)\hat{\pi}_n(dx_n|y_{1:n})$ converges in probability or almost surely to $\int \varphi(x)\pi_n(dx_n|y_{1:n})$. The proof works by induction on n, assuming that $\hat{\pi}_{n-1}$ is close to π_{n-1}. This error propagates in the next particle filter iteration, but one can control by how much it grows in the worst case, and the additional Monte Carlo error in the n-th step can be bounded by standard methods, at least with multinomial (independent) resampling. For balanced sampling, there seems to be still no general proof.

However, such a result is of limited use because the required sample size N may grow exponentially with the number of steps n. For applications, it is more relevant to find conditions under which the convergence is uniform in n. This is more difficult because – in contrast to the propagation step – the update step is in general not contractive and the above induction argument does not succeed. One has instead to study the error propagation over several time steps. This is equivalent to the question if and how fast the filter forgets its initial distribution π_0 which has been studied extensively, see e.g. Atar (2011).

6. More general situations

6.1. Filtering with continuous time observations

Much of the probability literature on filtering considers both state and observation processes in continuous time. More precisely, (Y_t) is assumed to satisfy the following evolution equation

$$dY_t = h(X_t)\,dt + dB_t$$

where (B_t) is a multivariate Brownian motion. We again denote by π_t the conditional distribution of X_t given the σ-field generated by the observations $(Y_s, s \le t)$ (completed by all null sets). Note that (π_t) is a stochastic process which takes values in the set of probability measures on $(S, \mathcal{S})$. The evolution equation for (π_t) corresponding to the recursions (5)–(6) is a stochastic PDE, the Kushner–Stratonovich equation. A particle filter approximation consists of interacting particles (X_t^i) and associated weights (W_t^i): Within an interval of length δ they evolve independently, whereas at multiples of δ there is a resampling step like in the discrete case, see Xiong (2011) for more details.

6.2. Sampling from moving targets

Particle filtering algorithms have found many applications outside the state space framework. In these cases, the more general term sequential Monte Carlo is used. Assume we have a complicated target distribution π on $(S, \mathcal{S})$ which we cannot sample directly. In such a situation, a promising strategy consists of sampling recursively from a sequence $\pi_0, \pi_1, \ldots, \pi_T$ where π_0

is a simple distribution, $\pi_T = \pi$ is the target one is interested in, and π_n is close to π_{n-1}. One example is the posterior distribution of a parameter with a large number T of observations where π_n the posterior for the first n observations. In another example, the π_n's are tempered approximations of $\pi = \pi_T$:

$$\pi_n(dx) \propto \left(\frac{\pi_T(dx)}{\pi_0(dx)}\right)^{\phi_n} \pi_0(dx) \quad (0 = \phi_0 < \phi_1 < \cdots < \phi_T = 1).$$

Starting with a sample from π_0, one wants to recursively generate samples (X_n^i) from π_n by resampling, propagation and reweighting as in the particle filter. If K_n denotes the transition kernel in the n-th propagation step, then for reweighting we need the density of π_n with respect to $\int \pi_{n-1}(dx')K_n(\cdot|x')$ which is typically not available in closed form, unless we choose K_n such that it leaves π_{n-1} invariant. The idea in Del Moral, Doucet and Jasra (2006) which allows more flexibility for the choice of K_n is to consider the distributions $\pi_{n-1}(dx')K_n(dx|x')$ and $\pi_n(dx)L_n(dx'|x)$ on the product space $(S, \mathcal{S})^2$. Here L_n is an arbitrary kernel such that these two distributions are absolutely continuous. If (X_{n-1}^i) is a (weighted) sample from π_{n-1} and we draw X_n^i from $K_n(dx|X_{n-1}^i)$ independently for different i's, then (X_{n-1}^i, X_n^i) is a (weighted) sample from $\pi_{n-1}(dx')K_n(dx|x')$. We can convert this into a weighted sample from $\pi_n(dx)L_n(dx'|x)$ because the Radon–Nikodym density can be computed without integration. By marginalization, we finally obtain the desired weighted sample from π_n. In Del Moral, Doucet and Jasra (2006), the optimal choice of L_n for given K_n is determined.

6.3. Rare event simulation

Particle filters are also used in rare event simulation, see e.g. Del Moral and Garnier (2005). Assume (Z_t) is a Markov process with fixed starting point z_0, τ and ζ are two stopping times and we are interested in $\mathbb{P}(\tau < \zeta)$ which is small so simple Monte Carlo is inefficient. In a technique called "importance splitting" one introduces a sequence of stopping times $0 = \tau_0 < \tau_1 < \cdots < \tau_T = \tau$ and sets $X_n = Z_{\tau_n}$. Moreover, we introduce "observations" $Y_n = 1_{[\tau_n < \zeta]}$. Then for $y_1 = y_2 = \cdots = y_T = 1$, π_n is the conditional distribution of Z_{τ_n} given $\tau_n < \zeta$, and $p(y_{1:T})$ is the probability we would like to estimate. Hence, we can estimate this probability unbiasedly with a particle filter since it gives *unbiased* estimates of $p(y_{1:n})$. Because the observations are deterministic functions of the state, the resampling step simply duplicates the particles with $\tau_n < \zeta$ until the sample size is again N. Amrein and Künsch (2011) propose to control precision instead of computational effort. This means that in the n-th step we do not propagate a fixed number of particles and see how many of them satisfy $\tau_n < \zeta$, but rather propagate particles until a fixed number of them satisfies $\tau_n < \zeta$. One can still obtain an unbiased estimator, and in addition this can increase the efficiency of the algorithm. Also in other applications of the particle filter where all observations are available from the beginning, it can be worthwile to aim for a fixed precision instead of a fixed computational effort in each iteration, using for instance accept-reject methods instead of importance sampling (see Künsch (2005)).

References

Aanonsen, S.I., Nævdal, G., Oliver, D.S., Reynolds, A.C. and Vallès, B. (2009). Ensemble Kalman filter in reservoir engineering – a review. *SPE Journal* (*Society of Petroleum Engineers*) **14** 393–412.

Amrein, M. and Künsch, H.R. (2011). A variant of importance splitting for rare event estimation: Fixed number of successes. *ACM Transactions on Modeling and Computer Simulation* **21** Article No. 13.

Andrieu, C., Doucet, A. and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **72** 269–342. MR2758115

Atar, R. (2011). Exponential decay rate of the filter's dependence on the initial distribution. In *The Oxford Handbook of Nonlinear Filtering* (D. Crisan and B. Rozovskiĭ, eds.) 299–318. Oxford: Oxford Univ. Press. MR2884600

Bickel, P., Li, B. and Bengtsson, T. (2008). Sharp failure rates for the bootstrap particle filter in high dimensions. In *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh. Inst. Math. Stat. Collect.* **3** 318–329. Beachwood, OH: IMS. MR2459233

Bretó, C., He, D., Ionides, E.L. and King, A.A. (2009). Time series analysis via mechanistic models. *Ann. Appl. Stat.* **3** 319–348. MR2668710

Briers, M., Doucet, A. and Maskell, S. (2010). Smoothing algorithms for state-space models. *Ann. Inst. Statist. Math.* **62** 61–89. MR2577439

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

Cappé, O., Moulines, E. and Rydén, T. (2005). *Inference in Hidden Markov Models. Springer Series in Statistics*. New York: Springer. MR2159833

Cressie, N. and Wikle, C.K. (2011). *Statistics for Spatio-temporal Data. Wiley Series in Probability and Statistics*. Hoboken, NJ: Wiley. MR2848400

Crisan, D. (2001). Particle filters – a theoretical perspective. In *Sequential Monte Carlo Methods in Practice* (A. Doucet, N. de Freitas and N. Gordon, eds.). *Stat. Eng. Inf. Sci.* 17–41. New York: Springer. MR1847785

Crisan, D. and Rozovskiĭ, B. (2011). *Handbook on Nonlinear Filtering*. Oxford: Oxford Univ. Press.

Del Moral, P. (2004). *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications. Probability and Its Applications* (*New York*). New York: Springer. MR2044973

Del Moral, P., Doucet, A. and Jasra, A. (2006). Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **68** 411–436. MR2278333

Del Moral, P. and Garnier, J. (2005). Genealogical particle analysis of rare events. *Ann. Appl. Probab.* **15** 2496–2534. MR2187302

Doucet, A., de Freitas, N. and Gordon, N., eds. (2001). *Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science*. New York: Springer. MR1847783

Doucet, A. and Johansen, A.M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In *Handbook on Nonlinear Filtering* (D. Crisan and B. Rozovskiĭ, eds.) 656–704. Oxford: Oxford Univ. Press.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research* **99** 10143–10162.

Evensen, G. (2007). *Data Assimilation: The Ensemble Kalman Filter*. New York: Springer.

Fearnhead, P., Wyncoll, D. and Tawn, J. (2010). A sequential smoothing algorithm with linear computational cost. *Biometrika* **97** 447–464. MR2650750

Frei, M. and Künsch, H.R. (2012). Bridging the ensemble Kalman and particle filter. Preprint. Available at arXiv:1208.0463v1.

Gilks, W.R. and Berzuini, C. (2001). Following a moving target – Monte Carlo inference for dynamic Bayesian models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **63** 127–146. MR1811995

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

Handschin, J.E. and Mayne, D.Q. (1969). Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *Internat. J. Control* (1) **9** 547–559. MR0246490

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME Ser. D. J. Basic Engrg.* **82** 35–45.

Kalman, R.E. and Bucy, R.S. (1961). New results in linear filtering and prediction theory. *Trans. ASME Ser. D. J. Basic Engrg.* **83** 95–108. MR0234760

Künsch, H.R. (2001). State space and hidden Markov models. In *Complex Stochastic Systems* (*Eindhoven*, 1999) (O.E. Barndorff-Nielsen, D.R. Cox and C. Klüppelberg, eds.). *Monogr. Statist. Appl. Probab.* **87** 109–173. Boca Raton, FL: Chapman & Hall/CRC. MR1893412

Künsch, H.R. (2005). Recursive Monte Carlo filters: Algorithms and theoretical analysis. *Ann. Statist.* **33** 1983–2021. MR2211077

Lei, J. and Bickel, P. (2011). A moment matching ensemble filter for nonlinear non-Gaussian data assimilation. *Monthly Weather Review* **139** 3964–3973.

Liu, J.S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statist. Comput.* **6** 113–119.

Pitt, M.K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *J. Amer. Statist. Assoc.* **94** 590–599. MR1702328

Wilkinson, D.J. (2011). *Stochastic Modelling for Systems Biology*, 2nd ed. Boca Raton, FL: Chapman & Hall/CRC.

Xiong, J. (2011). Particle approximations to the filtering problem in continuous time. In *The Oxford Handbook of Nonlinear Filtering* (D. Crisan and B. Rozovskiĭ, eds.) 635–655. Oxford: Oxford Univ. Press. MR2884611