MONTE CARLO FILTERING AND SMOOTHING WITH APPLICATION TO TIME-VARYING SPECTRAL ESTIMATION

Arnaud Doucet, Simon J. Godsill

Signal Processing Laboratory
University of Cambridge
Cambridge CB2 1PZ, UK
{ad2,sjg}@eng.cam.ac.uk

Mike West

Institute of Statistics and Decision Sciences
Duke University
Durham NC USA 27708-0251
mw@isds.duke.edu

ABSTRACT

We develop methods for performing filtering and smoothing in non-linear non-Gaussian dynamical models. The methods rely on a particle cloud representation of the filtering distribution which evolves through time using importance sampling and resampling ideas. In particular, novel techniques are presented for generation of random realisations from the joint smoothing distribution and for MAP estimation of the state sequence. Realisations of the smoothing distribution are generated in a forward-backward procedure, while the MAP estimation procedure can be performed in a single forward pass of the Viterbi algorithm applied to a discretised version of the state space. An application to spectral estimation for time-varying autoregressions is described.

1. INTRODUCTION

Consider the standard Markovian state-space model

$$\begin{aligned} x_{t+1} &\sim f(x_{t+1}|x_t) && \text{State evolution density} \\ y_{t+1} &\sim g(y_{t+1}|x_{t+1}) && \text{Observation density} \end{aligned}$$

where $\{x_t\}$ are unobserved states of the system and $\{y_t\}$ are observations made over some time interval $t \in \{1, 2, ..., T\}$. $f(.|.)$ and $g(.|.)$ are pre-specified state evolution and observation densities which may be non-Gaussian and involve non-linear. We assume that both $f(.|.)$ and $g(.|.)$ can be evaluated for any states and observations x_t and y_t. x_t and y_t may both in general be vectors. The Markov assumptions lead to the following expression for the joint distribution of states and observations by the probability chain rule

$$p\left(x_{1:t}, y_{1:t}\right) = f(x_1)\prod_{i=2}^{t} f(x_i|x_{i-1})\prod_{i=1}^{t} g(y_i|x_i)$$

This work is partly supported by the Engineering and Physical Sciences Research Council, UK

where $f(x_1)$ is the distribution of the initial state. $x_{1:t} \triangleq (x_1, ..., x_t)$ and $y_{1:t} \triangleq (y_1, ..., y_t)$ denote collections of observations and states from time 1 to t.

We will be concerned with sequential estimation of the filtering distribution $p(x_t|y_{1:t})$ and with simulation/optimisation of the entire smoothing distribution $p\left(x_{1:t}|y_{1:t}\right) \propto p\left(x_{1:t}, y_{1:t}\right)$. Updating of the filtering distribution can be achieved in principle using the standard filtering recursions

$$p(x_{t+1}|y_{1:t}) = \int p(x_t|y_{1:t}) f(x_{t+1}|x_t) dx_t$$

$$p(x_{t+1}|y_{1:t+1}) = \frac{g(y_{t+1}|x_{t+1}) p(x_{t+1}|y_{1:t})}{p(y_{t+1}|y_{1:t})}$$

Likewise, smoothing can be performed recursively backwards in time using the smoothing formula

$$p(x_t|y_{1:T}) = \int p(x_{t+1}|y_{1:T}) \frac{p(x_t|y_{1:t}) f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} dx_{t+1}$$

In practice these computations can only be performed in closed form for linear Gaussian models using the Kalman filter-smoother and for finite state-space hidden Markov models. In other cases approximate numerical techniques must be employed, such as the extended Kalman filter, Gaussian sum methods and general numerical integration procedures. Here we focus on Monte Carlo *particle filters* [1], [2], [3], in which the filtering distribution is approximated with an empirical distribution formed from point masses

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1$$

where $\delta(.)$ is the Dirac delta function and $w_t^{(i)}$ is the weight attached to particle $x_t^{(i)}$. Particles at time t can now be updated efficiently to particles at time

$t+1$ using importance sampling and resampling methods, see [3] for a review of the current methodology. In this paper we suppose that filtering has been performed, generating particles and weights $\{x_t^{(i)}, w_t^{(i)}\}$ for $t = 1, 2, ...T$, $i = 1, 2, ..., N$ (details of the filtering can be found in the time-varying AR models application described later).

2. SMOOTHING WITH PARTICLE FILTERS

2.1. Sampling from the smoothing distribution

We first develop a simple and efficient method for generation of realisations from the entire smoothing density $p(x_{1:T}|y_{1:T})$ using the particulate approximation. Other approaches to smoothing with particle filters have by contrast been aimed at approximating the *marginal* smoothing distributions $p(x_t|y_{1:T})$, either using the two-filter formula [**?**] or forward filtering-backward smoothing [3]. We obtain realisations from the smoothing density using the following factorisation

$$p(x_{1:T}|y_{1:T}) = \prod_{t=1}^{T} p(x_t|x_{t+1:T}, y_{1:T})$$

where

$$\begin{aligned} p(x_t|x_{t+1:T}, y_{1:T}) &= p(x_t|x_{t+1}, y_{1:t}) \qquad (1) \\ &= \frac{p(x_t|y_{1:t}) f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} \end{aligned}$$

This formula can be use to generate states successively in the reverse-time direction, conditioning upon future states. Specifically, given a random sample $\widetilde{x}_{t+1:T}$ drawn from $p(x_{t+1:T}|y_{1:T})$, draw $\widetilde{x}_t$ from $p(x_t|\widetilde{x}_{t+1:T}, y_{1:T})$. $(\widetilde{x}_t, \widetilde{x}_{t+1:T})$ is then a random realisation of $p(x_{t:T}|y_{1:T})$. The random draw from $p(x_t|\widetilde{x}_{t+1:T}, y_{1:T})$ relies on the particle approximation to $p(x_t|y_{1:t})$, as follows. The approximation to $p(x_t|\widetilde{x}_{t+1:T}, y_{1:T})$ can be obtained from formula (1), using the filtered particles $x_t^{(i)}$ and attaching importance weights $w_{t|t+1}^{(i)} \propto w_t^{(i)} f(\widetilde{x}_{t+1}|x_t^{(i)})$ (Note that the denominator term $p(\widetilde{x}_{t+1}|y_{1:t})$ need not be computed since it is constant for a particular instance of $\widetilde{x}_{t+1}$ and $y_{1:t}$. By contrast the smoothing method of [3] requires a Monte Carlo evaluation of this term.). Sampling from the discrete distribution with weights $w_{t|t+1}^{(i)}$ then completes the draw for $\widetilde{x}_t$. The algorithm proceeds as follows

Algorithm 1: Sampling from $p(x_{1:T}|y_{1:T})$

- Choose $\widetilde{x}_T = x_T^{(i)}$ with probability $w_T^{(i)}$.
- For $t = T-1$ to 1
  - Calculate $w_{t|t+1}^{(i)} \propto w_t^{(i)} f(\widetilde{x}_{t+1}|x_t^{(i)})$ for $i = 1, ..., N$
  - Choose $\widetilde{x}_t = x_t^{(i)}$ with probability $w_{t|t+1}^{(i)}$

$\widetilde{\mathbf{x}}_{1:T} = [\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_T]$ is an approximate realisation of $p(x_{1:T}|y_{1:T})$. Further independent realisations are obtained by repeating this procedure as many times as required.

2.2. MAP sequence estimation

The MAP estimate of state sequence is defined as

$$\widehat{x}_{1:T} = \arg\max_{x_{1:T}} p(x_{1:T}|y_{1:T})$$

The forward filtering pass generates a particle cloud of discrete states $\{x_t^{(i)}\}$ for $i = 1, ..., N$, $t = 1, ..., T$. This can be viewed as a discrete particle approximation to the whole continuous state space, containing all the 'significant' state values from the posterior distribution. The transition probability between any two successive states is easily evaluated as $f(x_{t+1}^{(i)}|x_t^{(j)})$, as is the observation density as $g(y_t|x_t^{(i)})$. The standard Viterbi algorithm can thus be adapted to the problem as follows:

Algorithm 2: MAP state sequence estimation

- $\alpha_0^{(i)} = 1, \quad i = 1, ..., N.$
- For $t = 1$ to T
  - For $i = 1, ..., N$
    - $\alpha_t^{(i)}(j) = f(x_t^{(i)}|x_{t-1}^{(j)}) g(y_t|x_t^{(i)}) \alpha_{t-1}^{(j)}$.
    - $\widehat{j} = \arg\max_{j \in \{1,...,N\}} \alpha_t^{(i)}(j)$, $\alpha_t^{(i)} = \alpha_t^{(i)}\left(\widehat{j}\right)$.
    - $\widehat{x}_{0:t}^{(i)} = (\widehat{x}_{0:t-1}^{(\widehat{j})}, x_t^{(i)})$
- $\widehat{i} = \arg\max_{i \in \{1,...,N\}} \alpha_T^{(i)}, \quad \widehat{x}_{0:T} = \widehat{x}_{0:T}^{(i)}$

$\widehat{x}_{0:T}$ is an estimate of the MAP state sequence. Note that this procedure can be performed in a single forward pass through the data, in contrast with the forward-backward procedure developed in algorithm 1. The complexity is greater, however, requiring $\mathcal{O}(N^2 T)$ computations as compared with $\mathcal{O}(NT)$ for each realisation of algorithm 1.

3. APPLICATION TO TVAR MODELS

In order to illustrate the applicability and usefulness of these methods, we test them in a time-varying autoregressive (TVAR) model.

3.1. Model specifications

The signal process $\{z_t\}$ is generated in the standard fashion from a Gaussian distribution centred at the linear prediction from the previous time step

$$f(z_t|a_t, \sigma_{e_t}) = \mathcal{N}\left(\sum_{i=1}^{P} a_{t,i} z_{t-i}, \sigma_{e_t}^2\right)$$

Here $a_t = (a_{t,1}, a_{t,2}, ..., a_{t,P})$ is the P^{th} order AR coefficient vector and $\sigma_{e_t}^2$ is the innovation variance at time t. Note that both of these quantities are time-varying, so we must specify an evolution model for these too. The signal is assumed to be observed in independent Gaussian noise, i.e. $g(y_t|x_t, \sigma_{v_t}) = \mathcal{N}\left(x_t, \sigma_{v_t}^2\right)$, where once again $\sigma_{v_t}^2$ may be time-varying. For our simulations, a Gaussian random walk model is assumed for the log-standard deviations $\phi_{e_t} = \log(\sigma_{e_t})$ and $\phi_{v_t} = \log(\sigma_{v_t})$

$$f(\phi_{e_t}|\phi_{e_{t-1}}, \sigma_{\phi_e}^2) = \mathcal{N}\left(\phi_{e_{t-1}}, \sigma_{\phi_e}^2\right), \quad (2)$$

$$f(\phi_{v_t}|\phi_{v_{t-1}}, \sigma_{\phi_v}^2) = \mathcal{N}\left(\phi_{v_{t-1}}, \sigma_{\phi_v}^2\right) \quad (3)$$

The model now requires specification of the time variation in a_t itself. This is the main interest in our applications, as we wish to find a model which is physically meaningful for the application and easily interpretable. Possibly the simplest choice of all is a Gaussian random walk directly on the coefficients, such as:

Model 1: Random Coefficient

$$f(a_t|a_{t-1}, \sigma_a^2) = \mathcal{N}\left(a_{t-1}, \sigma_a^2 I\right) \quad (4)$$

More elaborate schemes of this sort are possible, such as a smoothed random walk involving AR coefficients from further in the past, or a non-identity covariance matrix, but none of these modifications make a significant difference to the computations required in our scheme. Models of this form are, however, not constrained to be stable, which is a desirable property of real physical systems. A sufficient condition for stability of a TVAR model is that the instantaneous poles, i.e. the roots of the polynomial $(1 - \sum_{i=1}^{P} a_{t,i} p^{-i})$, lie strictly within the unit circle. This is over-stringent for ensuring the stability of a TVAR model. However, we have found simulated data from models whose poles occasionally move outside the unit circle can exhibit over-volatile behaviour, and so we will wish to enforce the more stringent constraint. This can be achieved by modifying the random walk appropriately.

Model 2: Random Coefficient, constrained form

$$f(a_t|a_{t-1}, \sigma_a^2) \quad (5)$$

$$\propto \begin{cases} \mathcal{N}\left(a_{t-1}, \sigma_a^2 I\right), & \max\{|p|;\ 1 - \sum_{i=1}^{P} a_{t,i} p^{-i} = 0\} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

We will see, however, that this condition is complex to simulate; moreover it is not possible to evaluate the distribution owing to the undetermined constant of proportionality. This model also does not have a strong physical interpretation in applications that we are familiar with.

Instead it is preferred to work in a domain where stability is easily monitored and incorporated into the filter/smoother. It would be possible to achieve this by modelling the roots directly (see [4], [5] for the time-invariant case). However, there are unresolved issues here of dealing with complex roots which evolve into real roots and *vice versa*. These issues do not arise if one works in the reflection coefficient, or equivalently partial correlation (PARCOR) coefficient domain [6]. Instead each reflection coefficient must simply be constrained to the interval $(-1, +1)$ in order to ensure strict stability. The standard Levinson recursion is used to transform between a_t and the reflection coefficients ρ_t. Many models are possible for the time variation of ρ_t, including random walks based on beta distributions and inverse logit transformed normal, and all would be feasible in our sequential Monte Carlo framework. We have chosen a simple truncated normal random walk.

Model 3: Random PARCOR, constrained form

$$f(\rho_t|\rho_{t-1}, \sigma_a^2) \propto \begin{cases} \mathcal{N}\left(\rho_{t-1}, \sigma_a^2 I\right), & \max\{|\rho_{t,i}|\} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The majority of our simulations have made use of this model. The TV-PARCOR framework is appealing, not simply because of the ease of checking stability and evaluating the transition density, but also because a reflection coefficient model has a strong physical interpretation in certain systems, notably speech and other acoustic sounds which are generated through tube-like mechanisms.

The state space model is now fully specified. The unobserved state vector is $x_t = (z_t, a_t, \phi_{e_t}, \phi_{v_t})$ in models 1 or 2, and $x_t = (z_t, \rho_t, \phi_{e_t}, \phi_{v_t})$ in model 3. Hyperparameters σ_a^2, $\sigma_{\phi_e}^2$ and $\sigma_{\phi_v}^2$ are assumed pre-specified and fixed in all the simulations. The initial state probability is diffuse Gaussian over the stable region in each of the models.

3.2. Filtering and smoothing

The first step in analysing the data is to perform a complete forwards sweep of a Monte Carlo filtering algorithm to produce weighted particles $\{x_t^{(i)}, w_t^{(i)}\}$ for $t = 1, 2, ...T$, $i = 1, 2, ..., N$, drawn approximately according to $p(x_t|y_{1:t})$. We omit the precise details here owing to lack of space, but note that this step is now a relatively standard procedure. We have experimented with two versions of the Monte Carlo filter, adapted to our specific TVAR model: the SIS/SIR filter [3] and the *auxiliary particle filter* [7]. We observe very little empirical difference in performance and report results based upon the SIS/SIR filter. The importance function employed is the state transition density of the states.Where a constrained model is required (models 2 and 3) rejection sampling is used to generate particles in the feasible region. We marginalise z_t from the filtering distribution in order to lend the algorithm more statistical stability. z_t is then straightforwardly sampled from its full conditional distribution, with no modification to the accrued weights. Following the filtering pass, smoothing is then carried out using algorithms 1 and 2.

4. RESULTS

Results are first presented for a simulated fourth order AR model. The fixed hyperparameters are $\sigma_a = 0.3$, $\sigma_{\phi_e} = 0.01$ and $\sigma_{\phi_v} = 0$. In these simulations ϕ_{v_t} is also assumed fixed and known with a value of 0.1. The initial states are assigned independent, diffuse Gaussian priors with zero mean. $N = 200$ data points are generated synthetically from model 3, the TV-PARCOR model. $N = 1000$ particles were used. The results of the filtering and subsequent smoothing using algorithms 1 and 2 are given in figure . In the first column, the true simulated values of the reflection coefficients are plotted. In the second column, the estimated mean of the filtering distribution (computed as $\sum_{i=1}^{N} \rho_{t,j}^{(i)} w_t^{(i)}$) is plotted (solid line). The dotted lines show the 5th and 95th percentiles of the same filtering distribution. The true parameters values are clearly covered by the inter-percentile region, and the estimated mean has tracked the parameters quite well. The third column shows 10 realisations of the smoothing density using algorithm 1. The final column gives the estimated MAP values for the parameters. In all cases, and particularly for the first two reflection coefficients, the estimated values have tracked the true values quite impressively, considering the complexity of the problem.

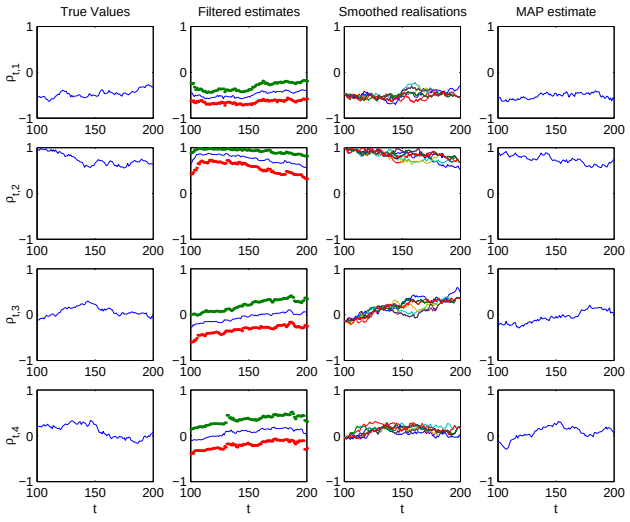

Figure 1: Results for a fourth order TV-PARCOR model - reflection coefficients.

REFERENCES

[1] N J Gordon, D J Salmond, and A F M Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, April 1993.

[2] J S Liu and R Chen. Sequential Monte Carlo methods for dynamical systems. *Journal of American Statistical Association*, 1998.

[3] A. Doucet, S.J. Godsill, and C. Andrieu. On sequential Monte Carlo methods for Bayesian filtering. *Statistics and Computing*, 1999. *(To appear).*

[4] G. Huerta and M. West. Bayesian inference on periodicities and component spectral structure in time series. *Journal of Time Series Analysis*, 20(4):401–416, 1999.

[5] G. Huerta and M. West. Priors and component structures in autoregressive time series models. *Journal of the Royal Statistical Society, Series B*, 1999.

[6] B. Friedlander. Lattice filters for adaptive processing. *Proc. IEEE*, 70(8):829–867, August 1982.

[7] M K Pitt and N Shephard. Filtering via simulation: auxiliary particle filters. *Journal of American Statistical Association*, 1998.