Sequential Monte Carlo Methods in State Space Models
Accessible Overview

Harry Spearing

February 19, 2018

We are a stationary observer and wish to track the position of an aircraft, but can only record the bearing of the aircraft, and not its position or speed. Is it possible to know the exact position? No, but it is still very possible to estimate this, and only two pieces of information are required. We need to know something about the relationship between:

1. any two successive measurements of the bearings.
2. a measurement of aircraft's bearing, and its corresponding speed or position.

The more we know about these relationships, the greater accuracy we have in estimating the full hidden data. Of course, if we knew the relationships exactly, all the information could be determined with exact precision. This is an example of a state space model, or hidden Markov

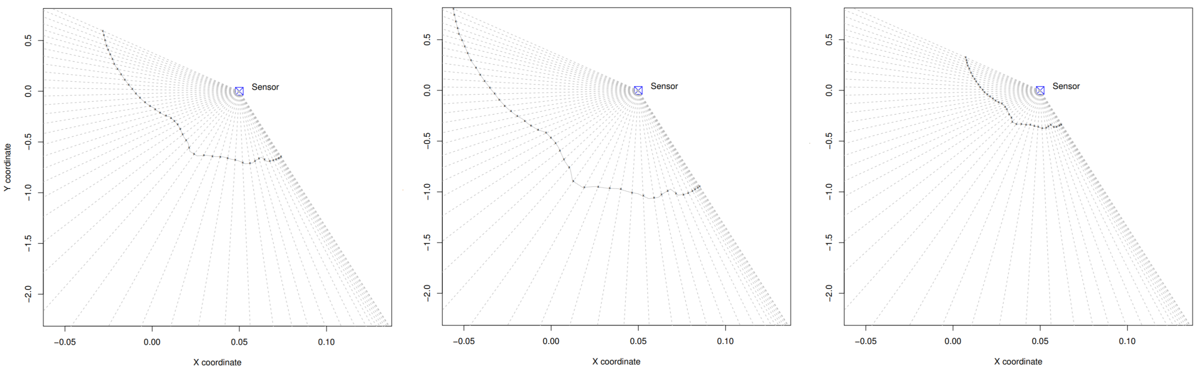

Figure 1: Tracking an aircraft using bearings only. Given a vector of bearings, any one of the above trajectories appear to be likely candidates. A particle filter can be applied here to estimate the full position of the aircraft.

model, and it has many other applications. More generally, we would like information about a system evolving in time, but without having direct access to all the data. The aim of a particle filter is to use approximate techniques to estimate this hidden data, based on the data we observe.

The Kalman filter presents an optimal estimate for the hidden states, but can only be applied in a very specific model. It is not strictly a particle filter, because it does not use approximate techniques. Instead the Kalman filter gives an analytical solution the this specific branch of hidden Markov models.

This report formulates the mathematics behind hidden Markov models, before introducing and simulating a Kalman filter. Later, approximate techniques are presented and applied to a hidden Markov model, with a simulation of one particular example, the Bootstrap filter. Throughout this report it is assumed that the parameters in the models are known, however in practice this is not the case.

Sequential Monte Carlo Methods in State Space Models

Harry Spearing

STOR601 Research Topic 1

Lancaster University

February 19, 2018

1 Introduction & Motivation

A State Space model is a partially observed Markov process. Also known as a Hidden Markov Model (HMM), a state space model provides an accurate framework for many applications, including macroeconomics [1] and facial recognition [2]. In human gait analysis for example, the observable quantities, such as arm and leg motion, are only partial observations of the full body movements which we wish to characterize. Another example is radar [5] where our states are the location and velocity of a target, but we can only make observations about the speed, or we can make observations about the velocity and angular component of the location, and wish to make inference about the radial distance from a point.
A particle filter is a sequential Monte Carlo method applied to a state space model. The aim of a particle filter is to estimate the full present latent (hidden) state, given the present and past partial observations. Smoothing is a similar technique, but uses all previous, present and future observations up to some later time to determine the present latent state. Trajectory estimates are smoother and more accurate than from filtering because more information is available, however smoothing can only be applied to off-line applications since it uses future data and as such, smoothing has less applications than filtering.
In some special cases, the best estimation for the latent state (filtered density) has a closed analytical form, and the Kalman filter[3] finds the optimum estimate. Most of the time however, the filtered density will need to be estimated using a Monte Carlo approach. There are many algorithms developed for this stochastic process including the Bootstrap Filter [4], which is introduced in this paper.

2 Formulation

The notation in the following sections will be defined as follows:

- $X_{1:T} = \{X_1, ..., X_T\}$ denote the hidden or *latent* processes we wish to estimate.
- $Y_{1:T} = \{Y_1, ..., Y_T\}$ are our set of observables.
- F_t and G_t are the transition and observation matrices respectively.

Our latent states follow a discrete-time Markov chain, that is

$$P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_0 = x_0) = P(X_t = x_t | X_{t-1} = x_{t-1}) \tag{1}$$

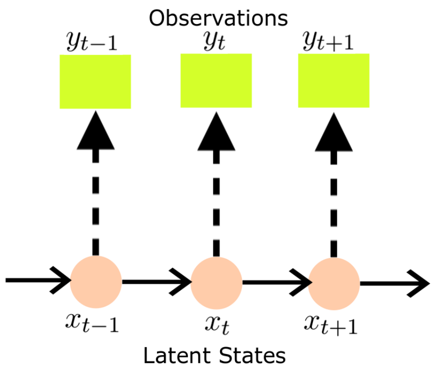

Figure 2: The latent states follow a Markov chain, and the observations are independent conditioned on the latent states.

We assume that the partial observations at time t, y_t are independent, conditioned on the corresponding latent state x_t.
Our interest is in estimating the filtered density.

$$p(x_t|y_{1:t}) = \frac{p(y_t, x_t|y_{1:t})}{p(y_t)} = \frac{p(y_t, x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{2}$$

Given some prior $\pi(x_1)$ about the initial latent state and a likelihood about the transition process, a posterior density of the latent states up to some time t' can be constructed.

$$p(x_{1:t'}, y_{1:t'}) = \pi(x_1) \prod_{t=2}^{t'} f(x_t|x_{t-1}) \prod_{t=1}^{t'} g(y_t|x_t) \tag{3}$$

Here, $f(\cdot)$ describes the transition probability density, and $g(\cdot)$ is the observation probability density function. Using Bayes rule, the filtered density (2) can be written as

$$p(x_t|y_{1:t}) = \int \frac{g(y_t|x_t) f(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1})}{p(y_t|y_{1:t-1})} dx_{t-1} \tag{4}$$

The nature of $f(\cdot)$ and $g(\cdot)$ allow the filtered density to be calculated sequentially as the latent process evolves.

3 Kalman Filter

A normal linear state space model, or normal dynamic linear model (NDLM) is a special case of the general state space model. In the NDLM the Kalman filter provides a closed form solution for latent state estimates. In the NDLM we have,

$$y_t = Gx_t + V_t \tag{5}$$
$$x_t = Fx_{t-1} + W_t \tag{6}$$

where V_t and W_t provide Gaussian noise. Because the state space model is linear-Gaussian, we can assume that the filtered densities are Gaussian. Given we that we know

$$p(x_{t-1}|y_{t-1}) \sim \mathcal{N}(\mu_{t-1}, \Sigma_{t-1})$$

we can use Kalman filter algorithm to find the filtered density at time t which will also be Gaussian. Therefore, it is necessary to find μ_t and Σ_t only. This is the aim of the Kalman filter.
There are 2 main steps to the algorithm - prediction,

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

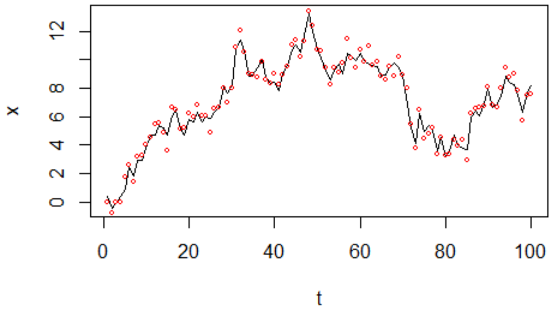

Figure 3: A simulation of Kalman recursions. The true latent values are shown by the red points, while the Kalman filter optimally estimates this based on the observed information.

and updating,

$$p(x_t|y_{1:t}) = \frac{g(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}$$

It can be shown [6] [7] that

$$\mu_t = \Sigma_t(C_t^{-1}F\mu_{t-1} + G^T W_t^{-1} y_t) \tag{7}$$

$$\Sigma_t^{-1} = C_t^{-1} + G^T W_t^{-1} G \tag{8}$$

where $C_t = V_t + F\Sigma_{t-1}F^T$.
These Kalman recursions were applied to the most simplest one dimensional linear Gaussian state space system shown in equation (9), the estimate of which is shown in figure 3

$$X_t \sim \mathcal{N}(X_{t-1}, 1), \quad Y_t \sim \mathcal{N}(X_t, 1), \quad X_0 \sim \mathcal{N}(0, 1) \tag{9}$$

4 Sequential Monte Carlo Methods

The Kalman filter optimally estimates the filtered density for linear Gaussian state space models and it can still produce good results when when the state space is near-linear, but when the model deviates from these conditions the Kalman filter performs poorly. By sequentially using Monte Carlo methods in discrete time, the filtered density can still be estimated. Sequentially applying a Monte Carlo estimate to state space models is known as a particle filter. In this section, we will cover sequential importance sampling (SIS) and the Bootstrap method, a sequential importance resampling (SIR) method.

4.1 Sequential Importance Sampler

Since the marginal density $p(y_t|y_{1:t-1})$ is not available in analytical form in all but the most simplest of scenarios, we wish to calculate this value up to a constant of proportionality. Equation 4 can approximated by sampling N independent random variables (particles) and assigning a weighting to each particle [8]. That is,

$$\hat{I} = \frac{\frac{1}{N}\sum_{i=1}^{N}\psi(x_i)\frac{p(x_i)}{q(x_i)}}{\frac{1}{N}\sum_{j=1}^{N}\frac{p(x_j)}{q(x_j)}}$$

$$\tilde{w}_j = \frac{p(x_i)}{q(x_i)}$$

Sequential Importance Sampling	
At time t=0	
For $i = 1, \dots, N$	
Sample	$x_{0,i} \sim p(x_0)$
Compute importance weights	$\tilde{w}_{0,i} = p(y_0 \vert x_0, i)$
	$w_{0,i} = \frac{\tilde{w}_{0,i}}{\sum_{j=1}^{N} \tilde{w}_{0,j}}$
At time t=T	
For $i = 1, \dots, N$	
Sample	$x_{t,i} \sim q(x_t \vert x_{t-1,i}, y_{0:t})$
	$x_{0:t,i} = (x_{0:t-1,i} x_{t,i})$
Compute importance weights	$\tilde{w}_{t,i} = \tilde{w}_{t-1,i} \frac{p(y_t \vert x_{t,i}) p(x_{t,i} \vert x_{t-1,i})}{q(x_{t,i} \vert x_{0:t-1,i}, y_{0:t})}$
Output latent states	$\{x_{t,i}, w_{t,i}\}_{i=1}^{N}$

Table 1: The SIS algorithm uses weighted samples at each iteration to estimate the latent states based on the observation distribution.

By defining the normalised weights as

$$w_i = \frac{\tilde{w}_i}{\sum \tilde{w}_j}$$

Here, $\psi(x)$ is an arbitrary function and $p(x)$ and $q(x)$ are probability densities such that the proposal distribution $q(x)$ has heavier tails than the target distribution $p(x)$, and therefore satisfies the heavy-tail rule [9] Therefore the integral can be approximated by $\hat{I} = \sum_{i=1}^{N} w_i \psi(x_i)$. Importance sampling offers way of estimating the filtered density without having to evaluate the marginal density. In order to do this, the choice of the proposal distribution $q(x)$ is important because this will effect the rate of convergence and therefore the accuracy of the approximation to the target distribution.
The sequential importance sampler (SIS) uses importance sampling at each iteration[10], the full algorithm is shown in table 4.2. Figure 4.1 helps to visualise the SIS algorithm being applied to a small number of particles. It appears to work well here, with the densest regions acquiring the greater weight and thus approximating the varying probabilities of the distribution $p(x)$. The problem occurs when the SIS algorithm is applied over many iterations because the variance of the importance weights increases over time. Essentially, the heavily weighted particles become even more heavily weighted, and the less weighted particles diminish further. There are few particles with non-negligible weight. This is a problem because our effective sample size N_{eff} decreases and so we have less and less information about the target distribution. This is known as the degeneracy problem of the Sequential Importance Sampler. There are three main ways of avoiding this degeneracy. Choosing a large N to start with means N_{eff} is still large enough by $t = T$. This will improve accuracy, however the computational cost increase with N. Secondly, choosing a proposal distribution $q(x)$ that is close to the target distribution $p(x)$ gives a more equal weighting to each of the particles at each iteration [11], as does using Markov chain Monte Carlo methods [12], however this will incur significant extra computational cost. Finally, resampling reduces the degeneracy.

4.2 Bootstrap Filter

By resampling at each iteration, we avoid the degeneracy issue of the SIS algorithm. The Bootstrap filter is an example of a Sequential Importance Resampler (SIR) with replacement algorithm [4]. Table 2 When the Bootstrap filter resamples, some Monte Carlo variance is introduced because the

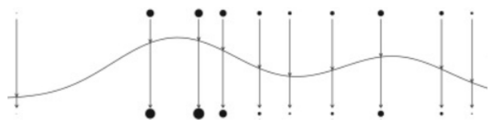

Figure 4: Illustration of the Sequential Importance Sampler applied to N=10 particles.

The Bootstrap filter.

At time t=0

For $i = 1, \ldots, N$
Sample $x_{0,i} \sim p(x_0)$
Compute importance weights $\tilde{w}_{0,i} = p(y_0|x_{0,i})$
$w_{0,i} = \frac{\tilde{w}_{0,i}}{\sum_{j=1}^{N} \tilde{w}_{0,j}}$

At time t=T

For $i = 1, \ldots, N$
Resample $\tilde{x}_{t-1,i}$ by resampling from $\{\tilde{x}_{t-1,i}\}_{i=1}^{N}$ with probability $\{w_{t-1,i}\}_{i=1}^{N}$
Sample $x_{t,i} \sim q(x_t|x_{t-1,i}, y_{0:t})$
$x_{0:t,i} = (x_{0:t-1,i} x_{t,i})$
Compute importance weights $\tilde{w}_{t,i} = \tilde{w}_{t-1,i} \frac{p(y_t|x_{t,i})p(x_{t,i}|x_{t-1,i})}{q(x_{t,i}|x_{0:t-1,i}, y_{0:t})}$

Output latent states $\{x_{t,i}, w_{t,i}\}_{i=1}^{N}$

Table 2: The Bootstrap filter also uses weighted samples, but then resamples particles with probability proportional to these weights to increase the effective sample size. After resampling, the weights of the new particles are set to be uniform.

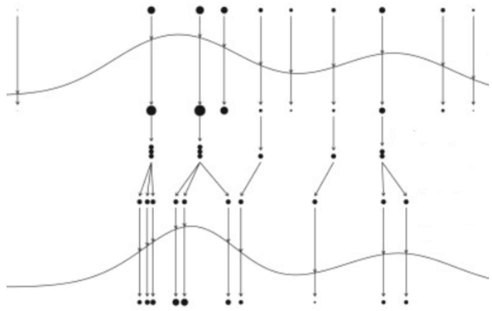

Figure 5: Illustration of the Bootstrap filter applied to N=10 particles.

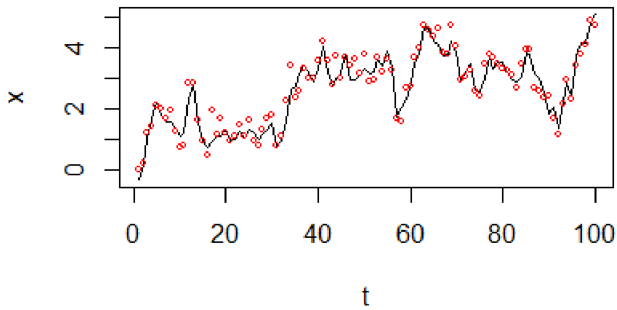

Figure 6: Simulation of a Bootstrap filter estimating the latent states of a NDLM. The true latent values are shown by the red points. The Bootstrap filter approximates this based on the observed information.

new set $\{\tilde{x}_{t,i}\}_{i=1}^{N}$ is only an approximation of the previous set $\{x_{t-1,i}\}_{i=1}^{N}$, so it would be sensible to resample only when necessary, that is, when the effective sample size of the new set is significantly smaller than the sample size of the previous set. N_eff can be estimated by [10]

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} w_{t,i}^2}$$

The resampled particles are resampled with a probability that is proportional to their weight. The new particles are then given equal weights, $w_{t,i} = \frac{1}{N}$.

The bootstrap filter was again applied to a one dimensional linear Gaussian state space system, shown in figure 4.2. It is not quite as accurate as the Kalman filter (which is optimum in this case) but does still appear to predict the latent states quite well.

5 Concluding Remarks

The Kalman filter predicts with optimal accuracy the latent states of linear Gaussian state space models, however when the model deviates from these conditions, it becomes increasingly inaccurate. We have seen that particle filters can be used to estimate the latent states of hidden Markov models by use of sequential Monte Carlo methods, where no such assumptions about linearity need be made. By resampling conditioned on the effective sample size, we can put a limit on the degeneracy incurred, whilst also limiting Monte Carlo variance.

In this paper it is assumed the parameters of the model are known, but in reality this is not the case, and parameter estimation must be considered. One common method for this is to use the Expectation-Maximisation algorithm [13].

References

[1] ERNNDEZ-VILLAVERDE, J. and RUBIO-RAMREZ, J. (2007). Estimating Macroeconomic Models: A Likelihood Approach. Review of Economic Studies, 74(4), pp.1059-1087.

[2] Mannini, A. and Sabatini, A. (2012). Gait phase detection and discrimination between walkingjogging activities using hidden Markov models applied to foot motion data from a gyroscope. Gait & Posture, 36(4), pp.657-661.

[3] Kalman, R. (1960). A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1), p.35.

[4] Gordon, N., Salmond, D. and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F Radar and Signal Processing, 140(2), p.107.

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

[6] Nemeth, C. (2014). Parameter estimation for state space models using sequential Monte Carlo algorithms. PhD, Lancaster University.

[7] Ziegel, E., West, M. and Harrison, J. (1997). Bayesian Forecasting and Dynamic Models. Technometrics, 39(4), p.433.

[8] Pollock, M. (2010). Introduction to Particle Filtering Discussion.

[9] Lampaki, I. (2015). Markov Chain Monte Carlo methodology for inference with generalised linear spatial models. PhD, Lancaster University.

[10] Kong A., Liu J.S. and Wong W.H. (1994) Sequential Imputations and Bayesian Missing Data Problems. Journal of the American Statistical Association, 89, 278-288.

[11] Pitt M.K. and Shephard N. (1999) Filtering via Simulation:Auxiliary Particle Filters. forthcoming Journal of the American Statistical Association.

[12] Berzuini C., Best N., Gilks W. and Larizza C. (1997) Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods. Journal of the American Statistical Association, 92, pp.1403-1412.

[13] Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. Biometrika, 98(1):6580.