An Introduction to Particle Filters

Paul Fearnhead
Lancaster University

September 12th, 2024

Aim of Session

- Introduce State Space Models.
- Derive the filtering equations for state space models.
- Show how the Bootstrap Filter can be used for inference.
- Understand the Monte Carlo error of the Bootstrap Filter.
- Discuss some ways of improving the basic Bootstrap Filter.
- Give pointers to the wider class of particle filters.

State Space Models

Particle filters allow for inference for state space models: models with

- an unobserved Markov process $X_1, X_2, \dots, X_t, \dots$; and
- partial observations $Y_1, Y_2, \dots, Y_t, \dots$, such that observation Y_t only depends on X_t (formally it is conditionally independent of the other X_s values given X_t).

We are interested in estimating the states, for example estimating X_t given $Y_{1:t} = \{Y_1, \dots, Y_t\}$.

Generally models will have unknown parameters but we will assume all parameters are known.

(Particle filters struggle with estimating parameters – in general you need to run them for different values.)

State Space Models

Mathematically we can define a state space model via:

$$X_1 \sim p\left(x_1\right),$$

$$X_t | X_{1:t-1} \sim p\left(x_t | x_{t-1}\right),$$

$$Y_t | X_{1:t}, Y_{1:t-1} \sim p\left(y_t | x_t\right),$$

where we use p for a general probability density (mass) function; and use the arguments to indicate which.

This leads to a simple way of simulating the state-space model.

In practice these densities may depend on parameters (but we are assuming these are known/conditioned on).

Filtering, Prediction and Smoothing

Particle filters are used to make inference about the latent states given the observations to date, $y_{1:t}$:

- Filtering: $p(x_t|y_{1:t})$. (Current state)
- Prediction: $p(x_T|y_{1:t})$, for $T > t$. (Future states)
- Smoothing: $p(x_T|y_{1:t})$, for $T < t$. (Past states)
- Inference: for the parameter θ, e.g. by calculating the likelihood, $p(y_{1:t}|\theta)$.

They are based on calculating these (particularly the filtering density) recursively.

Filtering Recursions

We can recursively calculate $p(x_t|y_{1:t})$ from $p(x_{t-1}|y_{1:t-1})$ and y_t by solving the filtering recursions:

$$
\begin{aligned}
p(x_t|y_{1:t-1}) &= \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})\mathrm{d}x_{t-1} && \text{(Prediction).} \\
p(x_t|y_{1:t}) &\propto p(x_t|y_{1:t-1})p(y_t|x_t) && \text{(Bayes' rule),} \\
p(y_t|y_{1:t-1}) &= \int p(x_t|y_{1:t-1})p(y_t|x_t)\mathrm{d}x_t && \text{(Normalising Constant)}
\end{aligned}
$$

These rely on the Markov property of the latent process and the conditional independence property of the observations.

Can be solved analytically for linear-Gaussian models (Kalman Filter).

Particle Filters

Particle Filters aim to approximately solve the filtering recursions using Monte Carlo. They are based on approximating the filtering densities by a weighted sample.

We will approximate $p(x_{t-1}|y_{1:t-1})$ by a set of particles

$$\{x_{t-1}^{(1)}, \ldots, x_{t-1}^{(N)}\},$$

with (normalised) weights

$$\{w_{t-1}^{(1)}, \ldots, w_{t-1}^{(N)}\}.$$

This means we approximate $p(x_{t-1}|y_{1:t-1})$ by a discrete distribution with value $x_{t-1}^{(i)}$ having probability $w_{t-1}^{(i)}$.

Particle Filters

How do we obtain $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ from $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^N$ (and y_t)?

The idea is to plug our approximation to $p(x_{t-1}|y_{1:t-1})$ into the filtering recursions.

This gives an approximation to $p(x_t|y_{1:t})$, and we then use a Monte Carlo method, normally Importance Sampling, to get weighted particles from this approximation.

Different Particle Filters differ in how they do the latter step.

Approximate Filtering Recursions

If we have weighted particles $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^N$ approximating $p(x_{t-1}|y_{t-1})$ we get the following approximate filtering recursions:

$$\hat{p}(x_t|y_{1:t-1}) = \sum_{i=1}^N w_{t-1}^{(i)} p(x_t|x_{t-1}^{(i)}) \qquad \text{(Prediction).}$$

$$\hat{p}(x_t|y_{1:t}) \propto \hat{p}(x_t|y_{1:t-1}) p(y_t|x_t) \qquad \text{(Bayes' rule),}$$

$$\hat{p}(y_t|y_{1:t}) = \int \hat{p}(x_t|y_{1:t-1}) p(y_t|x_t) \mathrm{d}x_t \qquad \text{(Normalising Constant)}$$

The approximate predictive distribution, $\hat{p}(x_t|y_{1:t-1})$, can be viewed as:
(i) sample $x_{t-1}^{(i)}$ from our discrete approximation to $p(x_{t-1}|y_{1:t-1})$, and
(ii) propagate this using the Markov dynamics of the latent state.

Bootstrap Filter

The idea of Importance Sampling is to get a weighted sample from the (approximate) filtering distribution, $\hat{p}(x_t|y_{1:t})$, is:

- We choose a proposal distribution, $q(x_t)$.
- We sample particles $x_t^{(i)}$ from $q(x_t)$, for $i = 1, \dots, N$.
- We calculate the (unnormalised) importance sampling weights

$$w_t^{*(i)} = \frac{\hat{p}(x_t^{(i)}|y_{1:t})}{q(x_t^{(i)})} \propto \frac{p(y_t|x_t^{(i)}) \sum_{j=1}^{N} w_{t-1}^{(j)} p(x_t^{(i)}|x_{t-1}^{(j)})}{q(x_t^{(i)})}.$$

- We normalise the weights

$$w_t^{(i)} = \frac{w_t^{*(i)}}{\sum_{i=1}^{N} w_t^{*(i)}}.$$

Bootstrap Filter

The Bootstrap filter (Gordon et al. 1993) implements this with

$$q(x_t) = \hat{p}(x_t|y_{1:t-1}) = \sum_{i=1}^{N} w_{t-1}^{(i)} p(x_t|x_{t-1}^{(i)}).$$

The importance sampling weight then simplifies to $p(y_t|x_t^{(i)})$.

This choice is natural, avoids the summation over the particles at time $t-1$ in the weight, and only requires one to be able to simulate from $p(x_t|x_{t-1})$ (rather than evaluate this density).

Bootstrap Filter

(i) Sample $x_t^{(i)}$, $i = 1, \dots, N$ from $\sum_{j=1}^N w_{t-1}^{(j)} p(x_t | x_{t-1}^{(j)})$, by: sampling j from $\{1, \dots, N\}$ with probabilities $\{w_{t-1}^{(k)}\}_{k=1}^N$, and then $x_t^{(i)}$ from $p(x_t | x_{t-1}^{(j)})$.

(ii) For $i = 1, \dots, N$ calculate the (unnormalised) weight

$$w^{*(i)} = p(y_t | x_t^{(i)}).$$

(iii) Normalise the weights, so for $i = 1, \dots, N$

$$w_t^{(i)} = \frac{w_t^{*(i)}}{\sum_{i=1}^N w_t^{*(i)}}.$$

And calculate the estimate of the conditional likelihood

$$\hat{p}(y_t | y_{1:t-1}) = \frac{1}{N} \sum_{i=1}^N w_t^{*(i)}.$$

Accuracy of Bootstrap/Particle Filter

There are two sources of error in our weighted particle approximation of $p(x_t|y_{1:t})$:

(i) The approximation of $p(x_{t-1}|y_{1:t-1})$, and how that propagates to the error of $\hat{p}(x_t|y_{1:t})$.

(ii) The additional Monte Carlo error from the importance sampling at time t.

The propagation of Error (i) depends on the model (how quickly it forgets the past); whereas Error (ii) depends on how we do the importance sampling.

Better particle filters try and reduce Error (ii).

Resampling/Stratified Sampling

Monte Carlo can be improved by using a stratified sample. This is primarily achieved through how we resample the particles from $t-1$:

- Any resampling is valid provided the mean number of times $x_{t-1}^{(i)}$ is resampled is $Nw_{t-1}^{(j)}$. This leads to residual and stratified resampling (e.g. Carpenter et al. 1999)
- The accuracy of one-step of the Bootstrap Filter can be measured through the Effective Sample size

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

  The accuracy of the importance sampling step is roughly the same as having ESS independent samples from $\hat{p}(x_t|y_{1:t})$.
  Often people propagate each $x_{t-1}^{(i)}$ (and weights) only if the ESS is large.

Auxillary Particle Filters

The Bootstrap filter performs poorly is the likelihood is peaked relative to the predictive distribution. (This can particularly be a problem at time $t = 1$; initialisation issues).

- We can use a better importance sampling proposal that uses the information in y_t.
- For $t > 1$, to avoid the $O(N)$ cost of calculating the importance sampling weight we use the auxillary particle filter of Pitt and Shephard (1999). (This proposes on the joint space of $(x_{t-1}^{(j)}, x_t)$.)
- Using the auxillary particle filter requires we can evaluate the transition density $p(x_t|x_{t-1})$.

Auxillary Particle Filters

We have a proposal distribution of the form $\sum_{j=1}^{N} \beta_{t-1}^{(j)} q(x_t|x_{t-1}^{(j)}, y_t)$. But simulate j with probability $\beta_{t-1}^{(j)}$ and x_t from $q(x_t|x_{t-1}^{(j)}, y_t)$.

The importance sampling weight for (j, x_t) is

$$\frac{w_{t-1}^{(j)} p(x_t|x_{t-1}^{(j)}, y_t)}{\beta_{t-1}^{(j)} q(x_t|x_{t-1}^{(j)}, y_t)}.$$

Optimally $\beta_{t-1}^{(j)} \propto w_{t-1}^{(j)} p(y_t|x_{t-1}^{(j)})$ and

$$q(x_t|x_{t-1}^{(j)}, y_t) \propto p(x_t|x_{t-1}^{(j)}) p(y_t|x_t),$$

when you are performing IID sampling from $\hat{p}(x_t|y_{1:t})$.

Parameter Estimation

- You can do parameter estimation by adding θ to the state. This works badly in general. (Adding MCMC moves to update θ can help; Fearnhead 2002).
- You can get an unbiased estimate of the likelihood $\hat{p}(y_{1:T})$ as

$$\log \hat{p}(y_{1:T}) = \log \hat{p}(y_1) + \sum_{t=2}^{T} \log \hat{p}(y_t | y_{1:t-1}).$$

  This can be used within pseudo-marginal MCMC (Andrieu and Roberts 2009).
- There are other ways of embedding particle filtering within MCMC (Andrieu et al. 2010)
- Some approaches try to estimate the gradient of the minus log-likelihood online and perform gradient descent (Kantas et al. 2015).

Smoothing

Smoothing is a more challenging problem. There are smoothing recursions that can be solved by particle filters (Kitagawa, 1996, Godsill et al. 2004).

Alternatively you can solve the smoothing problem by using a state $\tilde{X}_t = (X_1, \ldots, X_t)$. But this works badly.

Fixed lag-smoothing, with $\tilde{X}_t = (X_t - L + 1, \ldots, X_t)$ can work OK for small enough L.

References

Andrieu & Roberts (2009). *The pseudo-marginal approach for efficient Monte Carlo computations*. Annals of Statistics **37**, 697—725.

Andrieu et al. (2010). *Particle Markov chain Monte Carlo methods* JRSS B **72** 269-342.

Carpenter et al. (1999). *Improved particle filter for nonlinear problems*. IEE-F **146** 2–7

Chopin & Papaspiliopoulos (2020). *An introduction to Sequential Monte Carlo.*

Fearnhead (2002). *Markov chain Monte Carlo, sufficient statistics, and particle filters*. JCGS **11** 848–862.

Godsill et al. (2004). *Monte Carlo smoothing for nonlinear time series.* JASA **99**, 156–168.

Gordon et al. (1993). *Novel approach to nonlinear/non-Gaussian Bayesian state estimation* IEE-F **140**, 107–113

Kantas et al. (2015). *On Particle Methods for Parameter Estimation in State-Space Models*. Statistical Science, **30** 328–351.

Kitagawa (1996). *Monte Carlo filter and smoother for non-Gaussian nonlinear state space models* JCGS **5**, 1-25.

Pitt & Shephard (1999). *Filtering via simulation: auxiliary particle filters* JASA **94** 590–599