Sequential Monte Carlo Methods

Introduction

- MCMC methods are powerful, particular when Gibbs sampling methods can be used
- But they can be computationally demanding
- Gibbs sampling requires conditional posterior distributions to have forms that are easy to work with
- If posterior conditionals not known can be hard to get MCMC to work well
- Also typically need to program up new set of Gibbs sampling code for each model
- Sequential Monte Carlo methods are an alternative
- Suitable for any model, but most commonly used for nonlinear state space models
- Generic: write one set of code and, with minor alterations, can handle wide range of models

The Nonlinear State Space Model

- Let y_t for $t = 1, .., T$ be the observed data
- x_t be unobserved states (e.g. time varying parameters, volatilities, trend inflation, etc.)
- A generic nonlinear state space model involves $p\left(y_t|x_t\right)$ which defines the likelihood function

$$L = \prod_{t=1}^{T} p\left(y_t|x_t\right)$$

- and a prior for x_t taking the form

$$p\left(x_t|x_{t-1}\right)$$

- and an initial condition (which I will ignore for simplicity)

$$p\left(x_0\right)$$

- These densities may depend on other parameters but will suppress this to simplify notation

Example: Normal Linear State Space Model

- A TVP-VAR was written as:

$$y_t = Z_t \beta_t + \varepsilon_t$$

- With state equation:

$$\beta_{t+1} = \beta_t + u_t$$

- ε_t ind $N\left(0, \Sigma_t\right)$
- u_t ind $N\left(0, Q_t\right)$.
- ε_t and u_s are independent for all s and t.
- Link between this and notation used in this lecture:
- $x_t = \beta_t$
- $p\left(y_t | x_t\right)$ is $N\left(Z_t \beta_t, \Sigma_t\right)$
- $p\left(x_t | x_{t-1}\right)$ is $N\left(\beta_{t-1}, Q_t\right)$

Example: Stochastic Volatility

- Stochastic volatility model:

$$y_t = \exp\left(\frac{h_t}{2}\right)\varepsilon_t$$

- $$h_{t+1} = h_t + \eta_t$$

- ε_t is i.i.d. $N\left(0,1\right)$ and η_t is i.i.d. $N\left(0,\sigma_\eta^2\right)$. ε_t and η_s are independent.
- Link between this and notation used in this lecture:
- $x_t = h_t$
- $p\left(y_t \middle| x_t\right)$ is $N\left(0,\exp\left(h_t\right)\right)$
- $p\left(x_t \middle| x_{t-1}\right)$ is $N\left(h_{t-1},\sigma_\eta^2\right)$

Nonlinear State Space Models

- Many examples of nonlinear state space models in economics (e.g. DSGE models that have not been linearized)
- Typically MCMC methods are difficult unless a good approximation to the posterior can be found
- Sequential Monte Carlo methods may be a good alternative and have the advantage of being generic:
- Huge range of different models can be handled by simply adding code for your exact form for $p\left(y_t \middle| x_t\right)$ and $p\left(x_t \middle| x_{t-1}\right)$

Distributions of Interest

- Remember: notation convention where, e.g., $y^t = (y_1, ..., y_t)$
- Posterior distribution at time t: $p\left(x^t | y^t\right)$
- Filtering distribution at time t: $p\left(x_t | y^t\right)$
- Can be shown that, for this model:

$$\begin{array}{rcl} p\left(x^{t+1} | y^{t+1}\right) & \propto & p\left(x^t | y^t\right) p\left(y_{t+1} | x_{t+1}\right) p\left(x_{t+1} | x_t\right) \\ \left[\text{New Posterior}\right] & = & \left[\text{Old Posterior}\right] \left[\text{Term involving t+1 things}\right] \end{array}$$

- Where New = time t+1, Old = time t
- This is a recursive formula (and others below will also be)

Distributions of Interest

- Prediction distribution:

$$p\left(x_t|y^{t-1}\right) = \int p\left(x_t|x_{t-1}\right) p\left(x_{t-1}|y^{t-1}\right) dx_{t-1}$$

- Updating distribution:

$$p\left(x_t|y^t\right) \propto p\left(y_t|x_t\right) p\left(x_t|y^{t-1}\right)$$

- Same type of recursive structure
- For the Normal linear state space model all these densities have simple analytical forms and these are used in the algorithms described in Topic 5

Monte Carlo Integration and Importance Sampling

- Monte Carlo integration: take draws from posterior $p\left(x^t|y^t\right)$
- Can't be easily done for nonlinear state space models
- Importance sampling (similar to Metropolis-Hastings): take draws from a convenient importance function $q\left(x^t|y^t\right)$
- Take weighted average of importance sampling draws:

$$w_t = \frac{p\left(x^t|y^t\right)}{q\left(x^t|y^t\right)}$$

- Terminology: state space modellers often call such draws “particles", hence methods we will shortly describe sometimes called “particle filters”

Sequential Importance Sampling

- Suppose you are interested in sequence of posteriors $p\left(x^t|y^t\right)$ for $t=1,..,T$
- It is very convenient if, at each period in time, you do not have to redraw from the entire posterior
- That is, if $x^{t-1(s)}$ for $s=1,..,S$ are your draws at time $t-1$, much easier to retain them and just draw $x_t^{(s)}$ for the single x_t rather than draw entirely new $x^{t(s)}$ for $s=1,..,S$
- Can do this is if you choose your importance function to have the form:

$$q\left(x^t|y^t\right)=\prod_{k=1}^{t}q\left(x_k|x^{k-1},y^{k-1}\right)$$

- In this case, importance sampling weights used at time t, $w_t^{(s)}$ for $s=1,..,S$ are:

$$w_t^{(s)}\propto w_{t-1}^{(s)}\frac{p\left(y_t|x_t^{(s)}\right)p\left(x_t^{(s)}|x_{t-1}^{(s)}\right)}{q\left(x_t^{(s)}|x^{t-1(s)},y^{t-1}\right)}$$

Sequential Importance Sampling

- A simple choice for the importance function is:

$$q\left(x^t|y^t\right) = \prod_{k=1}^{t} p\left(x_k|x_{k-1}\right)$$

- In words: take draws from the state equation (prior)
- In this case, weights simplify to:

$$\begin{aligned} w_t^{(s)} &\propto w_{t-1}^{(s)} p\left(y_t|x_t^{(s)}\right) \\ \text{New weights} &= [\text{Old weights}] \times p\left(y_t|x_t^{(s)}\right) \end{aligned}$$

- Note simple recursive form
- This is the orginal version of the particle filter

Particle Filtering is Easy

- Remember: a nonlinear state space model is defined by $p\left(y_t|x_t\right)$ and $p\left(x_t|x_{t-1}\right)$
- Exact forms are application specific, but a key point is:
- ALL that strategy on previous slide involves is:
- Ability to take draws from $p\left(x_t|x_{t-1}\right)$ for $t = 1, .., T$
- Ability to evaluate $p\left(y_t|x_t\right)$
- Nothing else ... and all is done recursively so no need to redraw from entire posterior at each point in time

But Simple Particle Filtering Does Not Work too Well if T is Large

- But remember from our previous discussion of importance sampling:
- If importance function does not approximate posterior well, weights can become degenerate
- $w_t^{(s)} \approx 0$ for all but a few draws so effective number of draws is very small
- Unfortunately, this can often happen with sequential importance sampling if T is large
- Thus, there are many refinements on this simple particle filter to improve its performance
- For instance, some methods try to get better importance function than $p\left(x_t|x_{t-1}\right)$ while still retaining recursive structure
- No additional details given here for other types of Sequential Monte Carlo (SMC) methods
- See http://www.stats.ox.ac.uk/~doucet/smc_resources.html for state of the art

Informal Description of One Strategy Typically Used

- Problem is that importance sampling weights become degenerate as time passes
- There are measures of this degeneracy (e.g. Effective sample size = ESS)
- Strategy: If ESS falls too low, then do something
- Suppose some particles have low weight (bad particles) and others more weight (good particles)
- Bad particles in regions of low posterior probability, good particles in regions of high probability

- Bootstrap filter: get rid of bad particles and generate multiple values of good particles (with probabilities proportional to importance sampling weights)
- Get more good particles from region of high posterior probability
- This is called a resampling or a selection step
- Sometimes add mutation step where good particles are used to generate other nearby particles
- Key point: Once you have code that does particle filtering, possibly with resampling or mutation steps, it can handle any nonlinear state space model
- Just add your own code for drawing from $p\left(x_t|x_{t-1}\right)$ and evaluating $p\left(y_t|x_t\right)$
- MCMC methods usually much more case specific

What About Other Parameters?

- State space models often depend on other parameters, θ
- Defined in terms of $p\left(x_t | x_{t-1}, \theta\right)$ and $p\left(y_t | x_t, \theta\right)$
- If an estimate $\widehat{\theta}$ is available can proceed as above, using $p\left(x_t | x_{t-1}, \widehat{\theta}\right)$ and $p\left(y_t | x_t, \widehat{\theta}\right)$
- But there are methods being developed to combined inference on x_t and θ
- E.g. Chopin, Jacob and Papaspiliopoulos (2012, JRSS,B) develop SMC^2
- SMC on the states (as above) plus a method for SMC on the parameters
- Polson, Stroud and Muller (2008, JRSS,B) and Andrieu, Doucet and Holenstein (2010, JRSS,B) are other popular approaches

Summary

- Sequential Monte Carlo methods are not that common (yet) in economics
- Could be a very important tool in the future, especially for models where MCMC methods are hard to program up or computationally challenging
- They provide forecasts and marginal likelihoods recursively
- Do not have to re-run MCMC algorithm at each point in time
- Generic methods: once you have basic code produced, handling a new model is easy:
- Just add your own code for drawing from $p\left(x_t | x_{t-1}\right)$ and evaluating $p\left(y_t | x_t\right)$
- Can be parallelized so you can use the massive computing power in graphical processing units (GPUs)
- Open question: will they work well with high dimensional models like TVP-VARs or complicated nonlinear DSGE models?