Session 2: Introduction to Simulation-Based Inference

Bayesian Inference for the Social Sciences
November 3, 2017

Kevin Quinn

Why Use Simulation?

Why Use Simulation?

Recall that some quantities of interest are the following:

- The posterior mean of θ

$$\mathbb{E}[\theta|y] = \int_{\Theta} \theta p(\theta|y) d\theta$$

- The posterior variance of θ

$$\mathbb{V}[\theta|y] = \int_{\Theta} (\theta - \mathbb{E}[\theta|y])^2 p(\theta|y) d\theta$$

- a $100 \times (1-\alpha)\%$ credible set $C \subset \Theta$ where C is chosen to satisfy

$$1 - \alpha = \int_{C} p(\theta|y) d\theta$$

- the posterior predictive distribution:

$$p(y^{\text{rep}}|y) = \int_{\Theta} p(y^{\text{rep}}|\theta) p(\theta|y) d\theta$$

Why Use Simulation?

Recall that some quantities of interest are the following:

- The posterior mean of θ

$$\mathbb{E}[\theta|y] = \int_{\Theta} \theta p(\theta|y) d\theta$$

- The posterior variance of θ

$$\mathbb{V}[\theta|y] = \int_{\Theta} (\theta - \mathbb{E}[\theta|y])^2 p(\theta|y) d\theta$$

- a $100 \times (1-\alpha)\%$ credible set $C \subset \Theta$ where C is chosen to satisfy

$$1 - \alpha = \int_{C} p(\theta|y) d\theta$$

- the posterior predictive distribution:

$$p(y^{\text{rep}}|y) = \int_{\Theta} p(y^{\text{rep}}|\theta) p(\theta|y) d\theta$$

What do these quantities have in common?

Why Use Simulation?

The basic idea behind simulation-based inference is that we can calculate integrals like those above numerically rather than analytically.

Why Use Simulation?

The basic idea behind simulation-based inference is that we can calculate integrals like those above numerically rather than analytically.

Integrals of the form:

$$I = \int_{\Theta} g(\theta)p(\theta)d\theta$$

can be approximated numerically by taking a random sample $\{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(M)}\}$ of size M from the distribution with density $p(\theta)$ and calculating:

$$\widehat{I}_M = \frac{1}{M}\sum_{i=1}^{M} g(\theta^{(i)})$$

This is called Monte Carlo integration.

Basic Monte Carlo Integration

Classic Monte Carlo integration (Metropolis and Ulam, 1949) works by generating an independent sample from p.

Independence allows us to apply the Law of Large Numbers and show that $\widehat{I}_M$ converges to I as $M \to \infty$.

In other words, $\widehat{I}_M$ is a simulation consistent estimator of I.

Example 1: Voter ID Requests in Boston

Cobb, Greiner, and Quinn (2012) conducted an exit poll in Boston in 2008 and calculate the rates at which voters report being asked for ID at the polling location.

The exit poll covered 39 polling locations, but to make life simple we'll look at data from a single polling location. (We'll also ignore some complications caused by survey non-response, etc.)

In this polling location 314 voters were sampled. (The selection probability was 1/16 = 0.0625. 86 reported being asked for ID. What should we infer about the fraction of all (roughly 5000) voters in this polling location who were asked for ID?

Example 1a: Voter ID Requests in Boston

BostonVoters1.R

Example 1b: Voter ID Requests in Boston

`BostonVoters2.R`

Example 2: Authorship of the Disputed *Federalist Papers*

85 essays:

- 5 by John Jay
- 15 known to be by James Madison
- 51 known to be by Alexander Hamilton
- 3 known to be co-authored by Madison and Hamilton
- 11 of disputed authorship

Example 2: Authorship of the Disputed *Federalist Papers*

data: “stop word” frequencies in the essays known to be authored by only Hamilton or Madison

We invoke the bag of words assumption: conditional on the author-specific propensity to use each word, word usage is statistical independent within and across an author’s documents.

Let $\mathbf{y}_h$ denote the observed vector of stopwords frequencies from the known Hamilton essays and $\mathbf{y}_m$ denote the observed vector of stopwords frequencies from the known Madison essays.

Example 2: Authorship of the Disputed *Federalist Papers*

We assume that each of these word frequency vectors follows an author-specfic multinomial distribution:

$$p(\mathbf{y}_h|\boldsymbol{\theta}_h) = \frac{n!}{y_{h1}!y_{h2}!\cdots y_{hk}!}\theta_{h1}^{y_{h1}}\theta_{h2}^{y_{h2}}\cdots\theta_{hk}^{y_{hk}}$$

and

$$p(\mathbf{y}_m|\boldsymbol{\theta}_m) = \frac{n!}{y_{m1}!y_{m2}!\cdots y_{mk}!}\theta_{m1}^{y_{m1}}\theta_{m2}^{y_{m2}}\cdots\theta_{mk}^{y_{mk}}$$

Example 2: Authorship of the Disputed *Federalist Papers*

We assume that each of these word frequency vectors follows an author-specfic multinomial distribution:

$$p(\mathbf{y}_h|\boldsymbol{\theta}_h) = \frac{n!}{y_{h1}!y_{h2}!\cdots y_{hk}!}\theta_{h1}^{y_{h1}}\theta_{h2}^{y_{h2}}\cdots\theta_{hk}^{y_{hk}}$$

and

$$p(\mathbf{y}_m|\boldsymbol{\theta}_m) = \frac{n!}{y_{m1}!y_{m2}!\cdots y_{mk}!}\theta_{m1}^{y_{m1}}\theta_{m2}^{y_{m2}}\cdots\theta_{mk}^{y_{mk}}$$

Our prior is that $\boldsymbol{\theta}_h$ and $\boldsymbol{\theta}_m$ follow independent Dirichlet distributions:

$$p(\boldsymbol{\theta}_h) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_k)}\theta_{h1}^{(\alpha_1-1)}\cdots\theta_{hk}^{(\alpha_k-1)}$$

and

$$p(\boldsymbol{\theta}_m) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_k)}\theta_{m1}^{(\alpha_1-1)}\cdots\theta_{mk}^{(\alpha_k-1)}$$

Example 2: Authorship of the Disputed *Federalist Papers*

Ignoring constants of proportionality we can write:

$$
\begin{aligned}
p(\boldsymbol{\theta}_h|\mathbf{y}_h) &\propto p(\mathbf{y}_h|\boldsymbol{\theta}_h)p(\boldsymbol{\theta}_h) \\
&= \left\{\theta_{h1}^{y_{h1}}\theta_{h2}^{y_{h2}}\cdots\theta_{hk}^{y_{hk}}\right\}\left\{\theta_{h1}^{(\alpha_1-1)}\theta_{h2}^{(\alpha_2-1)}\cdots\theta_{hk}^{(\alpha_k-1)}\right\} \\
&= \theta_{h1}^{(y_{h1}+\alpha_1-1)}\theta_{h2}^{(y_{h2}+\alpha_2-1)}\cdots\theta_{hk}^{(y_{hk}+\alpha_k-1)}
\end{aligned}
$$

which is proportional to a Dirichlet density.

$p(\boldsymbol{\theta}_m|\mathbf{y}_m)$ can be calculated analogously.

Example 2: Authorship of the Disputed *Federalist Papers*

Let $\mathbf{z}$ denote the observed vector of word frequencies for one of the disputed essays.

Further suppose that our prior belief is that there is a 50% chance this essay was authored by Hamilton and a 50% chance it was authored by Madison.

Example 2: Authorship of the Disputed *Federalist Papers*

Let $\mathbf{z}$ denote the observed vector of word frequencies for one of the disputed essays.

Further suppose that our prior belief is that there is a 50% chance this essay was authored by Hamilton and a 50% chance it was authored by Madison.

If Hamilton was the author, then $\mathbf{z}$ was generated from a multinomial distribution with parameter $\boldsymbol{\theta}_h$.

If Madison was the author, then $\mathbf{z}$ was generated from a multinomial distribution with parameter $\boldsymbol{\theta}_m$.

Example 2: Authorship of the Disputed *Federalist Papers*

Let $\mathbf{z}$ denote the observed vector of word frequencies for one of the disputed essays.

Further suppose that our prior belief is that there is a 50% chance this essay was authored by Hamilton and a 50% chance it was authored by Madison.

If Hamilton was the author, then $\mathbf{z}$ was generated from a multinomial distribution with parameter $\boldsymbol{\theta}_h$.

If Madison was the author, then $\mathbf{z}$ was generated from a multinomial distribution with parameter $\boldsymbol{\theta}_m$.

Using Bayes rule (and an abuse of notation):

$$\Pr(Hamilton|\mathbf{z}, \boldsymbol{\theta}_h, \boldsymbol{\theta}_m) = \frac{0.5 \times p(\mathbf{z}|\boldsymbol{\theta}_h)}{0.5 \times p(\mathbf{z}|\boldsymbol{\theta}_h) + 0.5 \times p(\mathbf{z}|\boldsymbol{\theta}_m)}$$

Example 2: Authorship of the Disputed *Federalist Papers*

`FP-simple-analysis.R`

Basic Monte Carlo Integration

Basic Monte Carlo integration works great when you can easily take an independent sample from the distribution of interest.

However, in many realistic examples it is not feasible to generate an independent sample from the posterior distribution.

What can be done in these situations?

Markov Chain Monte Carlo (MCMC)

Idea behind MCMC is that rather than trying to generate an independent sample from $p(\theta|y)$ we are going to construct a dependent sample approximately from $p(\theta|y)$ using a Markov chain

This sounds difficult but is usually quite easy

By utilizing particular types of Markov chains we can still construct simulation consistent estimators of many integrals even though the usual Law of Large Numbers doesn't hold anymore

The Metropolis-Hastings Algorithm

Suppose we want to sample from $p(\theta|y)$, but we only know what this is up to a constant of proportionality

The Metropolis-Hastings Algorithm

Suppose we want to sample from $p(\theta|y)$, but we only know what this is up to a constant of proportionality

We can't sample from $p(\theta|y)$ directly

The Metropolis-Hastings Algorithm

Suppose we want to sample from $p(\theta|y)$, but we only know what this is up to a constant of proportionality

We can't sample from $p(\theta|y)$ directly

Can we sample approximately from $p(\theta|y)$ using only the information in the unnormalized function $h(\theta|y) = p(y|\theta)p(\theta) \propto p(\theta|y)$?

The Metropolis-Hastings Algorithm

Suppose we want to sample from $p(\theta|y)$, but we only know what this is up to a constant of proportionality

We can't sample from $p(\theta|y)$ directly

Can we sample approximately from $p(\theta|y)$ using only the information in the unnormalized function $h(\theta|y) = p(y|\theta)p(\theta) \propto p(\theta|y)$?

Yes– the Metropolis-Hastings algorithm allows us to do just this

The Metropolis-Hastings Algorithm

The general Metropolis-Hastings algorithm is:

initialize $\theta^{(0)}$
for (i in 1 to M){
  sample $\theta_{can}^{(i-1)}$ from $q(\theta_{can}|\theta^{(i-1)})$
  set

$$\theta^{(i)} = \begin{cases} \theta_{can}^{(i-1)} & \text{with probability } \alpha(\theta_{can}^{(i-1)}, \theta^{(i-1)}) \\ \theta^{(i-1)} & \text{with probability } 1 - \alpha(\theta_{can}^{(i-1)}, \theta^{(i-1)}) \end{cases}$$

store $\theta^{(i)}$
}
where

$$\alpha(\theta_{can}, \theta) = \min\left\{ \frac{p(\theta_{can}|y)}{p(\theta|y)} \; \frac{q(\theta|\theta_{can})}{q(\theta_{can}|\theta)}, 1 \right\}$$

The Metropolis-Hastings Algorithm

Note: the algorithm depends on the ratios $p(\theta_{can}|y)/p(\theta|y)$ and $q(\theta|\theta_{can})/q(\theta_{can}|\theta)$

The Metropolis-Hastings Algorithm

Note: the algorithm depends on the ratios $p(\theta_{can}|y)/p(\theta|y)$ and $q(\theta|\theta_{can})/q(\theta_{can}|\theta)$

- We can substitute $h(\theta|y)$ for $p(\theta|y)$ and we similarly only need to know the candidate generating density up to a constant of proportionality (that doesn't depend on θ)

The Metropolis-Hastings Algorithm

Note: the algorithm depends on the ratios $p(\theta_{can}|y)/p(\theta|y)$ and $q(\theta|\theta_{can})/q(\theta_{can}|\theta)$

- We can substitute $h(\theta|y)$ for $p(\theta|y)$ and we similarly only need to know the candidate generating density up to a constant of proportionality (that doesn't depend on θ)

Different variations of the Metropolis-Hastings algorithm can be created depending on the choice of candidate generating density $q(\theta_{can}|\theta)$

Random-Walk Metropolis

Assume candidate values are generated according to $\theta_{can}^{(i-1)} = \theta^{(i-1)} + \epsilon$ where ϵ is assumed to follow a symmetric density f

Random-Walk Metropolis

Assume candidate values are generated according to $\theta_{can}^{(i-1)} = \theta^{(i-1)} + \epsilon$ where ϵ is assumed to follow a symmetric density f

This means the candidate generating density will be $q(\theta_{can}|\theta) = f(\theta_{can}^{(i-1)} - \theta^{(i-1)})$

Random-Walk Metropolis

Assume candidate values are generated according to $\theta_{can}^{(i-1)} = \theta^{(i-1)} + \epsilon$ where ϵ is assumed to follow a symmetric density f

This means the candidate generating density will be $q(\theta_{can}|\theta) = f(\theta_{can}^{(i-1)} - \theta^{(i-1)})$

By symmetry of f, $q(\theta_{can}|\theta) = q(\theta|\theta_{can})$

Random-Walk Metropolis

Assume candidate values are generated according to $\theta_{can}^{(i-1)} = \theta^{(i-1)} + \epsilon$ where ϵ is assumed to follow a symmetric density f

This means the candidate generating density will be $q(\theta_{can}|\theta) = f(\theta_{can}^{(i-1)} - \theta^{(i-1)})$

By symmetry of f, $q(\theta_{can}|\theta) = q(\theta|\theta_{can})$

This means the acceptance probability is just $\alpha(\theta_{can}, \theta) = \min\{p(\theta_{can}|y)/p(\theta|y),\ 1\}$

Random-Walk Metropolis

Assume candidate values are generated according to $\theta_{can}^{(i-1)} = \theta^{(i-1)} + \epsilon$ where ϵ is assumed to follow a symmetric density f

This means the candidate generating density will be $q(\theta_{can}|\theta) = f(\theta_{can}^{(i-1)} - \theta^{(i-1)})$

By symmetry of f, $q(\theta_{can}|\theta) = q(\theta|\theta_{can})$

This means the acceptance probability is just $\alpha(\theta_{can}, \theta) = \min\{p(\theta_{can}|y)/p(\theta|y),\ 1\}$

Easy to set up and get running

Random-Walk Metropolis

Assume candidate values are generated according to $\theta_{can}^{(i-1)} = \theta^{(i-1)} + \epsilon$ where ϵ is assumed to follow a symmetric density f

This means the candidate generating density will be $q(\theta_{can}|\theta) = f(\theta_{can}^{(i-1)} - \theta^{(i-1)})$

By symmetry of f, $q(\theta_{can}|\theta) = q(\theta|\theta_{can})$

This means the acceptance probability is just $\alpha(\theta_{can}, \theta) = \min\{p(\theta_{can}|y)/p(\theta|y),\ 1\}$

Easy to set up and get running

May not take full advantage of what you know about a problem

Example 4: Voter ID Requests in Boston (Again)

`BostonVoters1-RWMH.R`

Independent Metropolis-Hastings

Suppose candidate values are generated from a density q that does not depend on the current value of θ

Independent Metropolis-Hastings

Suppose candidate values are generated from a density q that does not depend on the current value of θ

Acceptance probabilities are:

$$\alpha(\theta_{can}, \theta) = \min\left\{ \frac{p(\theta_{can}|y)}{p(\theta|y)} \; \frac{q(\theta)}{q(\theta_{can})}, 1 \right\}$$

Independent Metropolis-Hastings

Suppose candidate values are generated from a density q that does not depend on the current value of θ

Acceptance probabilities are:

$$\alpha(\theta_{can}, \theta) = \min\left\{\frac{p(\theta_{can}|y)}{p(\theta|y)} \frac{q(\theta)}{q(\theta_{can})}, 1\right\}$$

Is usually either extremely efficient or extremely inefficient

Independent Metropolis-Hastings

Suppose candidate values are generated from a density q that does not depend on the current value of θ

Acceptance probabilities are:

$$\alpha(\theta_{can}, \theta) = \min\left\{\frac{p(\theta_{can}|y)}{p(\theta|y)}\ \frac{q(\theta)}{q(\theta_{can})}, 1\right\}$$

Is usually either extremely efficient or extremely inefficient

Depends on how close q is to the target density p

Independent Metropolis-Hastings

Suppose candidate values are generated from a density q that does not depend on the current value of θ

Acceptance probabilities are:

$$\alpha(\theta_{can}, \theta) = \min\left\{\frac{p(\theta_{can}|y)}{p(\theta|y)}\ \frac{q(\theta)}{q(\theta_{can})}, 1\right\}$$

Is usually either extremely efficient or extremely inefficient

Depends on how close q is to the target density p

As a practical matter, the chain will behave reasonably only if the tails of q are heavier than the tails of p

Independent Metropolis-Hastings

Suppose candidate values are generated from a density q that does not depend on the current value of θ

Acceptance probabilities are:

$$\alpha(\theta_{can}, \theta) = \min\left\{\frac{p(\theta_{can}|y)}{p(\theta|y)}\ \frac{q(\theta)}{q(\theta_{can})}, 1\right\}$$

Is usually either extremely efficient or extremely inefficient

Depends on how close q is to the target density p

As a practical matter, the chain will behave reasonably only if the tails of q are heavier than the tails of p

"Tailored M-H" (Chib and Greenberg): find the MLEs and asymptotic variance-covariance matrix– use these to construct a multivariate-*t* candidate generating density

Example 5: Voter ID Requests in Boston (Again)

`BostonVoters1-IndMH.R`

Practical Issues with MCMC: MH Acceptance Rates

Need to monitor the acceptance rate (fraction of candidates accepted)

Practical Issues with MCMC: MH Acceptance Rates

Need to monitor the acceptance rate (fraction of candidates accepted)

Acceptance rate too high– high autocorrelation in the chain– poor mixing– not moving quickly around the parameter space

Practical Issues with MCMC: MH Acceptance Rates

Need to monitor the acceptance rate (fraction of candidates accepted)

Acceptance rate too high– high autocorrelation in the chain– poor mixing– not moving quickly around the parameter space

Acceptance rate too low– inefficient– most of the computational time is being spent generating draws that are ultimately rejected

Practical Issues with MCMC: MH Acceptance Rates

Need to monitor the acceptance rate (fraction of candidates accepted)

Acceptance rate too high– high autocorrelation in the chain– poor mixing– not moving quickly around the parameter space

Acceptance rate too low– inefficient– most of the computational time is being spent generating draws that are ultimately rejected

"Too high" and "too low" depend on the exact algorithm

Practical Issues with MCMC: MH Acceptance Rates

Need to monitor the acceptance rate (fraction of candidates accepted)

Acceptance rate too high– high autocorrelation in the chain– poor mixing– not moving quickly around the parameter space

Acceptance rate too low– inefficient– most of the computational time is being spent generating draws that are ultimately rejected

“Too high” and “too low” depend on the exact algorithm

- Random walk Metropolis: somewhere between 0.25 and 0.50 is usually recommended (inversely related to number of parameters)

Practical Issues with MCMC: MH Acceptance Rates

Need to monitor the acceptance rate (fraction of candidates accepted)

Acceptance rate too high– high autocorrelation in the chain– poor mixing– not moving quickly around the parameter space

Acceptance rate too low– inefficient– most of the computational time is being spent generating draws that are ultimately rejected

"Too high" and "too low" depend on the exact algorithm

- Random walk Metropolis: somewhere between 0.25 and 0.50 is usually recommended (inversely related to number of parameters)
- Independent M-H: something close to 1 is preferred (as long as you know the candidate generating density is close to p)

Practical Issues with MCMC: Mixing and Convergence

Since the stationary distribution is approached in the limit, the first samples from the chain are possibly from distributions that are quite different from the stationary distribution

Practical Issues with MCMC: Mixing and Convergence

Since the stationary distribution is approached in the limit, the first samples from the chain are possibly from distributions that are quite different from the stationary distribution

To eliminate sensitivity to the starting value of the chain it is customary to discard the first b samples as burn-in

Practical Issues with MCMC: Mixing and Convergence

Since the stationary distribution is approached in the limit, the first samples from the chain are possibly from distributions that are quite different from the stationary distribution

To eliminate sensitivity to the starting value of the chain it is customary to discard the first b samples as burn-in

As long as the starting values are from an area of the parameter space that has a reasonable amount of posterior mass burn-in can be quite short

Practical Issues with MCMC: Mixing and Convergence

Since the stationary distribution is approached in the limit, the first samples from the chain are possibly from distributions that are quite different from the stationary distribution

To eliminate sensitivity to the starting value of the chain it is customary to discard the first b samples as burn-in

As long as the starting values are from an area of the parameter space that has a reasonable amount of posterior mass burn-in can be quite short

A bigger concern is typically how well the chain is mixing or exploring the parameter space

Practical Issues with MCMC: Mixing and Convergence

Since the stationary distribution is approached in the limit, the first samples from the chain are possibly from distributions that are quite different from the stationary distribution

To eliminate sensitivity to the starting value of the chain it is customary to discard the first b samples as burn-in

As long as the starting values are from an area of the parameter space that has a reasonable amount of posterior mass burn-in can be quite short

A bigger concern is typically how well the chain is mixing or exploring the parameter space

If the chain is moving around the parameter space slowly then the chain needs to be run for a large number of iterations

Practical Issues with MCMC: Mixing and Convergence

In the early 1990s many practitioners recommended taking every kth draw from a chain to produce an approximately iid sample

Practical Issues with MCMC: Mixing and Convergence

In the early 1990s many practitioners recommended taking every kth draw from a chain to produce an approximately iid sample

This practice is known as thinning the chain

Practical Issues with MCMC: Mixing and Convergence

In the early 1990s many practitioners recommended taking every kth draw from a chain to produce an approximately iid sample

This practice is known as thinning the chain

It turns out that this isn't necessary for inference because of the ergodic theorem

Practical Issues with MCMC: Mixing and Convergence

In the early 1990s many practitioners recommended taking every kth draw from a chain to produce an approximately iid sample

This practice is known as thinning the chain

It turns out that this isn't necessary for inference because of the ergodic theorem

In fact, it has been shown that thinning always increases the variance of one's Monte Carlo estimates

Practical Issues with MCMC: Mixing and Convergence

In the early 1990s many practitioners recommended taking every kth draw from a chain to produce an approximately iid sample

This practice is known as thinning the chain

It turns out that this isn't necessary for inference because of the ergodic theorem

In fact, it has been shown that thinning always increases the variance of one's Monte Carlo estimates

Nonetheless, thinning does save RAM and hard disk space when working with models (such as ideal point models) with hundreds or thousands of parameters.

Practical Issues with MCMC: Mixing and Convergence

There is no way to prove that most MCMC algorithms have converged to the stationary distribution in a particular finite number of iterations

Numerous diagnostics exist, all have weaknesses

One long run or several short runs?

Practical Advice: look at a number of diagnostics but don't be a slave to a particular diagnostic

If in doubt, run the chain out longer