

Prof. H. Bensmail

Stat-664: Adv Stat Infer

Spring Semester 2006

**Chapter 11:**  
**Markov chain simulations**  
**April 12, 2006**

## 11.1 Introduction

Markov chain is used to simulate a random walk in the space of  $S$  which converges to a stationary distribution that is the joint posterior distribution,  $p(\theta|y)$

The key to Markov chain simulation is to create a Markov process whose stationary distribution is aspecified  $p(\theta|y)$  and run the simulation long enough that the distribution of the current draws is close enough to the stationary distribution.

A variety of Markov chains with  $p(\theta|y)$  stationary distribution can be constructed.

There is no fully satisfactory method for drawing simulations in general, but the following approach is often successful for simulating from the posterior distributions.

- 1.
2. create an approximate posterior density based on the joint or marginal modes  $P_{approx}$  and draw samples from the approximate distribution  $P_{approx}$  and use importance resampling, to sample about 10 or 20 draws of the parameter vector. Eliminates samples with low importance ratio and multiply samples with high importance ratio by the importance ratio.
3. using these as starting points, run independent parallel sequences of an iterative simulation such as the Gibbs sampler or Metropolis algorithm.
4. run the iterative simulation until approximate convergence appears to have been reached, means when

$$\sqrt{\hat{R}} = \sqrt{\frac{(n-1)W + \frac{1}{n}B}{W}} \text{ where } W = \text{within variance, and } B = \text{between variance}$$

5. if  $\sqrt{\hat{R}}$  is near 1 for all scalar estimands of interest, summarize inference about the posterior distribution by treating the set of all iterates from the second half of the simulated sequences as an identically distributed sample from the target function.
6. Compare the posterior inferences from the markov chain simulation to the approximate distribution used to start the simulations. If they are close with respect to locations and approximate distributional shape, check for the errors before believing that the markov chain simulation has produced a better answer.

## 11.2: The Metropolis algorithm

Here, the innovation is to construct and sample from a transition distribution  $T$  for arbitrary posterior distribution.

The Metropolis algorithm is a general term for a family of Markov chain simulation methods that are useful for simulation from posterior distributions.

So Given a target function

$$p(\theta|y)$$

Metropolis algorithm creates a sequence of random points

$$\theta^1, \theta^2, \dots, \theta^T$$

whose distribution converges to the target distribution.

The algorithm is described as the following:

1. Draw a starting point  $\theta^0$ , for which  $P(\theta^0|y) > 0$ .
2. For  $t = 1, \dots, T$ 
  - Sample a candidate point  $\theta^*$  from a jumping distribution  $J(\theta^*|\theta^{t-1})$  at time  $t$  (random walk)  
the jumping distribution must be symmetric, that is  $J(\theta_a|\theta_b) = J(\theta_b|\theta_a)$  for all  $\theta_a, \theta_b$  and  $t$   
Clearly a sufficient (but not necessary) condition for an equilibrium probability distribution is the so-called detailed balance condition which means that

$$p(\theta_a|y) \times J(\theta_a|\theta_b) = p(\theta_b|y) \times J(\theta_b|\theta_a)$$

Nevertheless, all these conditions do not determine the transition probability uniquely.

A simple choice for  $J$  fulfilling the above conditions is given in terms of the energy change as proposed by Metropolis

$$\exp\{-\beta U(\theta)/z\}$$

from the physics theory

- Calculate the acceptance ratio

$$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$

- set

$$\theta^t = \left\{ \begin{array}{l} \theta \text{ with probability } \min(r, 1) \\ \theta \text{ otherwise} \end{array} \right\}$$

which means: draw

$$U \sim \text{uniform}(0, 1)$$

accept  $\theta^*$  if  $U < r$

reject else

- The algorithm requires the ability to calculate the relative importance ratio,  $r$  for all  $(\theta, \theta^*)$  and to draw  $\theta$  from the jumping distribution  $J_t(\theta^*|\theta)$  for all  $\theta$  and  $t$ .

### Example 11.1:

Suppose that the target function is

$$p(\theta|y) = N_2(0, I)$$

The jumping distribution is

$$J(\theta^*|\theta^{t-1}) = N_2(\theta^{t-1}, \left(\frac{1}{5}\right)^2 I)$$

So at the iteration  $t$ , we simulate  $\theta^*$  from  $J(\theta^*|\theta^{t-1})$

The ratio at each step is

$$\begin{aligned} r &= \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)} \\ &= \frac{N_2(\theta^*|0, I)}{N_2(\theta^{t-1}|0, I)} \end{aligned}$$

It is hard to give general advice on efficient jumping rules, but for normal distributions for example, we have the following:

$$\theta = (\theta_1, \theta_2, \dots, \theta_p)$$

suppose that  $\theta$  has a multinormal distribution with known variance  $\Sigma$

Then we propose a symmetric normal jumping kernel with the same shape as the target function  $p(\theta|y)$

$$J(\theta^t|\theta^{t-1}) = N_p(\theta^{t-1}, c^2\Sigma)$$

among this class of jumping rules, the most efficient has scale

$$c = 2.5/\sqrt{p}$$

### Example 11.2:

Suppose we have a joint density function

$$\begin{aligned} p(\theta, \mu, \log \sigma, \log \tau|y) &\propto \tau \prod_j N(\theta_j|\mu, \tau^2) \\ &\times \prod_j \prod_i^{n_j} N(y_{ij}|\theta_j, \sigma^2) \end{aligned}$$

Then we first calculate the mode of  $(\theta_j, \mu, \sigma, \tau)$  using the conditional maximization

$$p(\theta_j|\mu, \sigma, \tau, y) \sim N(\hat{\theta}_j, V_{\theta_j})$$

$$p(\mu|\theta, \sigma, \tau, y) \sim N(\hat{\mu}, \tau^2)$$

$$p(\sigma^2|\theta, \mu, \tau, y) \sim IG\left(\frac{n}{2}, \frac{n\hat{\sigma}^2}{2}\right)$$

$$p(\tau^2|\theta, \mu, \sigma, \tau, y) \sim IG\left(\frac{J-1}{2}, \frac{(J-1)\hat{\tau}^2}{2}\right)$$

also we can calculate the marginal posterior mode of  $p(\mu, \log \sigma, \log \tau|y)$  using EM algorithm where:

E-step: calculate the expectation of

$$p(\theta, \mu, \log \sigma, \log \tau|y)$$

over  $\theta$

M-Step: Maximize  $E[p(\theta, \mu, \log \sigma, \log \tau|y)]$  as a function of  $(\mu, \log \sigma, \log \tau)$  and to find the mode  $(\hat{\mu}, \log \hat{\sigma}, \log \hat{\tau})$

and then we can use a normal approximation centered at the mode using a  $(3 \times 3)$

second derivative matrix of the marginal posterior density  $p(\mu, \log \sigma, \log \tau | y)$

$$p(\mu, \log \sigma, \log \tau | y)$$

we can draw  $(\mu, \log \sigma, \log \tau)$  from the normal approximation

we can draw  $\theta$  from  $p(\theta | \mu, \log \sigma, \log \tau, y)$  which is already normally distributed

Here we can improve the simulation using Metropolis algorithm for example:

1. we sample  $t = 200$  draws from the normal approximation for  $p(\mu, \log \sigma, \log \tau | y)$ , then we draw a subsample of size 10 using importance resampling and use these as starting points
2. use the metropolis to jump through the marginal posterior distribution  $p(\mu, \log \sigma, \log \tau | y)$ . We jump through the space of  $(\mu, \log \sigma, \log \tau)$ , using a multivariate normal jumping kernel with variance matrix equal to that of the normal approximation multiplied by  $c^2 = (2.5/\sqrt{3})^2$
3. draw simulations of the vector  $\theta$  from its normal conditional posterior distribution.

### 11.3: The Gibbs sampler algorithm

For the many multidimensional problems, Gibbs sampler is very useful. It is an alternating conditional sampling. Each iteration of the Gibbs cycles through the subvectors of  $\theta$ , drawing each subset conditional on the value of all the others.

Suppose that

$$\theta = (\theta_1, \theta_2, \dots, \theta_p)$$

then there are  $p$  steps in one iteration  $t$ . In fact we iterate conditional sampling as the following:

1. draw  $\theta_j^{(t)} \sim$  conditional distribution  $p(\theta_j | \theta_{-j}^{(t-1)}, y)$ , where

$$\theta_{-j} = (\theta_1^{(t)}, \theta_2^{(t)}, \dots, \theta_{j-1}^{(t)}, \theta_{j+1}^{(t-1)}, \dots, \theta_p^{(t-1)})$$

Thus each value is updated conditional on the latest value of  $\theta$  for the other components which are the iteration  $t$  values for the components already updated and the iteration  $(t - 1)$

values for the others.

#### Example 11.3:

suppose the target distribution is:

$$\begin{pmatrix} X \\ Y \end{pmatrix} | \rho \sim N \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right]$$

then the marginal distributions

$$(X | Y = y) \sim N(\rho y, 1 - \rho^2)$$

$$(Y | X = x) \sim N(\rho x, 1 - \rho^2)$$

Then the Gibbs sampler steps are :

1. start from, say  $(X, Y) = (10, 10)$ , we can take a look at the trajectories. If  $\rho = 0.6$ , then the first 20 iterations are

*Let's do this for 100, 1000, 10,000 iterations*

```
*****
> traject
function()
{
x=rep(0,200)
y=rep(0,200)
x[1]=10
y[1]=10
for(i in 2:200){
x[i]=rmnorm(1,0.6*y[i-1],1-(0.6)^2)
y[i]=rmnorm(1,0.6*x[i-1],1-(0.6)^2)
}
par(mfrow= c(2,1))
ts.plot(x)
ts.plot(y)
}
*****
```

#### **Example 11.4:**

Now if the target distribution

$$f(x, y) \propto \binom{n}{x} y^{x+\alpha-1} (1-y)^{n-x+\beta-1}$$

$$x = 0, \dots, n; 0 \leq y \leq 1$$

the marginal are:

$$(x|y) \sim ?$$

$$y|x \sim ?$$

How do we draw samples?

1. draw  $x$  from :
2. draw  $y$  from
3. for illustrations, let's take  $\alpha = 0.5$ , and  $n = 10$

```
*****
> traject2
function()
{
y<-rep(0,10000)
r<-rep(0,10000)
y[1]<-0.5
12
}
```

```
r[1]<-5n<-10for(iin2:10000){
x<-runif(n)
z<-(x<y[i-1])
r[i]<-sum(z)
y[i]<-rbeta(1,r[i-1]+ 0.5,n-r[i-1]+ 0.5)
}
par(mfrow= c(2,2))
hist(r)
hist(y)
tsplot(r)
tsplot(y)
}
*****
```

### Example 11.5