Chapter 8

Markov Chain Simulations

8.1 Introduction

The early examples of bayesian analysis where we had to use simulation describe simulation approaches that work in low-dimensional problems (grid approach in bivariate or univariate cases when we did not know the shape of the distributions, or simulating directly from a known distribution). The theory behind those methods was the Montecarlo method which says that if you draw L random numbers

$$\theta_i \sim p(\theta \mid y)$$

the θ_i are independent and identically distributed and then the average of any function of θ can be approximated by the average of those random numbers, i.e.,

$$E(h) = \int h(\theta) p(\theta \mid y) d\theta \approx \frac{1}{L} \sum h(\theta_i).$$

With complicated models, it is rare that samples from the posterior distribution can be obtained directly. This chapter talks about the method of Markov chain simulation, in particular, Gibbs sampling and Metropolis Hasting algorithms. With these methods, it is not important that the θ_i are *iid*.

The key to Markov chain simulation is to create a Markov process whose stationary distribution is a specified $p(\theta \mid y)$. They rely on the possibility of producing (with a computer) an endless flow of random variables for known or new distributions. Such a simulation is, in turn, based on the production of uniform random variables on the interval (0,1).

8.2 Markov Chain Monte Carlo to summarize posterior distributions

MCMC algorithms are very attractive in that they are easy to set up and program and require relatively little prior input from the user. R is a convenient language for programming these algorithms and is also very suitable for performing output analysis, where one does several graphical and numerical computations to check if the the algorithm is indeed producing draws from the target posterior distribution.

8.2.1 Introduction to discrete markov chains

A markov chain describes probabilistic movements between several states. A person starts at location i of locations 1, 2, 3, 4, 5, 6. The probability that the person moves to another location j depends on the current location i only.

Def: Transition probabilities: describe the likelihood of moving between particular states in one step.

Def: A transition matrix T summarizes the transition probabilities.

$$T = \begin{bmatrix} 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ 0.4 & 0.5 & 0.1 & 0 & 0 & 0 \\ 0.2 & 0.5 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0.25 & 0.5 & 0 \\ 0 & 0 & 0.25 & 0.35 & 0.4 & 0 \\ 0 & 0 & 0 & 0.25 & 0.5 & 0.25 \end{bmatrix}$$

The first row in T gives the probabilities of moving to all states 1 through 6 in a single step from location 1, the second row gives the transition probabilities in a single step from location 2, and so on.

Def: Irreducible transition matrix: it is possible to go from every state to every state in one or more steps.

Def: Periodic transition matrix: given that a person is in state i, if the person can only return to this state at regular intervals. Our transition matrix is aperiodic.

We represent one's current location as a probability row vector of the form

$$p = (p_1, p_2, p_3, p_4, p_5, p_6)$$

where p_i represents the probability the person is currently in state i. If p^j represents the location of the person at state j, then the location of the person at state $j + 1$ is

$$p^{j+1} = p^j T$$

Suppose we can find a probability vector w such that $wT = w$. Then w is said to be the stationary distribution. If a Markov chain is irreducible and aperiodic, it has a unique stationary distribution. Moreover, the limiting distribution of this Markov chain, as the number of steps approaches infinity, will be equal to this stationary distribution.

We empirically demonstrate the existence of the stationary distribution of our Markov chain by running a simulation experiment. We start our simulation in R by reading in the transition matrix and setting up a storage vector s for the location of the traveler in the random walk.

```
T = matrix(c(0.7, 0.3,0,0,  0,  0, 0.4, 0.5,0.1,0, 0,0, 0.2, 0.5, 0.3,
0, 0, 0, 0 ,0,0.25, 0.25, 0.5, 0, 0, 0, 0.25, 0.35, 0.4, 0, 0, 0, 0,
0.25, 0.5, 0.25  ), ncol=6, byrow=T)
T
s=array(0,c(50000,1))
```

Assume that the starting location for our traveler is state 3. We will write a loop to simulate 50,000 draws from the Markov chain. We use the `sample` function to simulate one step -the arguments to this function indicate that we are sampling a single value from the set {1,2,3,4,5,6} with probabilities given by the s^{j-1} row of the transition matrix T, where s^{j-1} is the current location for our traveler.

```
  s[1]=4
for (j in 2:50000)
s[j]=sample(1:6,size=1,prob=T[s[j-1], ]   )
```

We summarize the frequencies of visits to the six states after 500, 2000, 8000, 50000) steps of the chain by use of the `table` command; we convert the counts to relative frequencies by dividing by the number of steps.

```
m=c(500,2000,8000,50000)
 for(i in 1:4)
print(table(s[1:m[i]])/m[i])
```

It appears from the output that the relative frequencies of the states are converging to the stationary distribution $w = (0.1, 0.2, 0.2, 0.2, 0.2, 0.1)$. We can confirm that w is indeed the stationary distribution of this chain by multiplying w by the transition matrix T:

```
w=matrix(c(0.1,0.2,0.2,0.2,0.2,0.1), nrow=1,ncol=6)
w %*% T
```

which gives w

8.2.2 Markov Chain Montecarlo

MCMC methods is a continuous-valued generalization of the discrete Markov chain setup described above. The MCMC sampling strategy sets up an irreducible, aperiodic Markov chain for which the stationary distribution equals the posterior distribution of interest. A general way of constructing a Markov chain is by the Metropolis-Hasting algorithm. The independence chain and the random walk chain are two particular variants of the MH algorithm that are applicable to a wide variety of Bayesian inference problems.

The purpose of Markov chain Monte Carlo (MCMC) techniques is to replace a difficult analytical integration process with iterative work by the computer. When calculations on posterior distributions are multidimentional, there is a need to summarize each marginal distribution to provide useful results to readers in a regresion-style table or other format. The basic principle behind MCMC techniques is that if an iterative chain of computer-generated values can be set up carefully enough, and run long enough, then *empirical* estimates of integral quantities of interest can be obtained from summarizing the observed output. If each visited multidimensional location is recorded as a row vector in a matrix, then the marginalization for some parameter of interest is obtained simply by summarizing the individual dimension down the corresponding column. So we replace an analytical problem with a sampling problem, where the sampling process has the computer perform the difficult and repetitive processes. This is an enormously important idea to Bayesians and to others since it frees researchers from having to make artificial simplifications to their model specifications just to obtain describable results.

These Markov chains are successive quantities that depend probabilistically only on the value of their immediate predecessor: the *Markovian property*. In general, it is possible to set up a chain to estimate multidimensional probability structures (i.e., desired probability distributions), by starting a Markov chain in the appropriate sample space and letting it run until it settles into the target distribution. Then when it runs for some time confined to this particular distribution, we can collect summary statistics such as means, variances and quantiles from their simulated values. This idea has revolutionized Bayesian statistics by allowing the empirical estimation of probability distributions that could not be analytically calculated.

8.2.3 Simple Gibbs sampling

This section introduces an important, and frequently used Markov chain Montecarlo tool, the Gibbs sampler. The idea behind a Gibbs sampler is to get a marginal distribution for each variable by iteratively conditioning on interim values of the others in a continuing cycle until samples from this process empirically approximate the desired marginal distribution. Standard regression tables that appear in journals are simply marginal descriptions. There will be much more on this topic throughout the course, but here we will implement a simple but instructive example.

Suppose that we have two conditional distributions:

$$f(\theta_1 \mid \theta_2) \propto \theta_2 \exp -\theta_2\theta_1, \qquad f(\theta_2 \mid \theta_1) \propto \exp -\theta_1\theta_2, \qquad 0 < \theta_1, \theta_2 < B < \infty$$

These conditional distributions are both exponential probability density functions. The upper bound, B, is important since without it there is no finite joint density and the Gibbs sampler will not work. It is possible, but not particularly pleasant, to perform the correct integration steps to obtain the desired marginal distributions: $f(x)$ and $f(y)$. Instead, we will let the Gibbs sampler do the work.

The Gibbs sampler is a "transition kernel defined by full conditional distributions" that allows us to run a Markov chain that eventually settles into the desired limiting distribution that characterizes the marginals. In plainer language, it is an iterative process that cycles through conditional distributions until it reaches a stable status whereby future samples characterize the desired distributions. The important theorem here assures us that when we reach this stable distribution, the autocorrelated sequence of values can be treated as an iid sample from the marginal distributions of

interest. The amazing part is that this is accomplished simply by ignoring the time index, i.e. putting the values in a bag and just "shaking it up." Gibbs sampling is actually even more general than this.

For two parameters, θ_1 and θ_2, this process involves a starting point, $[\theta_1^{(0)}, \theta_2^{(0)}]$, and the cycles are defined by drawing random values from the conditionals according to:

$$\begin{array}{rr}
\theta_1^{(1)} \sim f(\theta_1 \mid \theta_2^{(0)}) & \theta_2^{(1)} \sim f(\theta_2 \mid \theta_1^{(1)}) \\
\theta_1^{(2)} \sim f(\theta_1 \mid \theta_2^{(1)}) & \theta_2^{(2)} \sim f(\theta_2 \mid \theta_1^{(2)}) \\
\theta_1^{(3)} \sim f(\theta_1 \mid \theta_2^{(2)}) & \theta_2^{3} \sim f(\theta_2 \mid \theta_1^{(1)}) \\
\cdots & \cdots \\
\cdots & \cdots \\
\theta_1^{(m)} \sim f(\theta_1 \mid \theta_2^{(m-1)}) & \theta_2^{(m)} \sim f(\theta_2 \mid \theta_1^{(m)})
\end{array}$$

If we are successful, then after some reasonable period the values θ_1, θ_2 are safely assumed to be empirical samples from the correct marginal distribution. There are many theoretical and practical concerns that we are ignoring here, and the immediate objective here is to give a rough overview.

8.2.4 Example: Bivariate normal distribution (Gelman et al. p 327)

Consider a single observation (y_1, y_2) from a bivariate normally distributed population with unknown mean (θ_1, θ_2) and known covariance matrix $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$. With a uniform prior distribution on θ, the posterior distribution is

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \mid y \sim N\left(\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right)$$

Although we don't need it, consider the Gibbs sample for the purpose of exposition. To apply the Gibbs sampler to (θ_1, θ_2), we need the conditional posterior distributions, which, from the properties of the multivariate normal distribution, are also normal.

$$\theta_1 \mid \theta_2, y \sim N(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2) \tag{8.1}$$
$$\theta_2 \mid \theta_1, y \sim N(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2) \tag{8.2}$$

The Gibbs sampler proceeds by alternately sampling from these two normal distributions. Suppose $\rho = 0.8$, data $(y_1, y_2) = (0, 0)$. Thus, we will be simulating from the following two conditionals:

$$lll\theta_1 \mid \theta_2 \sim N(0.8\theta_2, 0.2) \tag{8.3}$$
$$\theta_2 \mid \theta_1 \sim N(0.8\theta_1, 0.2) \tag{8.4}$$

Let's start a sequence with initial values $\theta_1^0 = 2.5$

```
####### Gibbs sampling #############
theta1=rep(0,5000)
theta2=rep(0,5000)
theta2[1]=2.5
theta1[1]= rnorm(1,0.8*theta2[1], 0.2)
for(i in 2:5000){
   theta2[i] = rnorm(1,0.8*theta1[i-1], 0.2)
   theta1[i]=rnorm(1,0.8*theta2[i],0.2)
```

```
   }

######## Summarize the marginals #######
#### after removing first 500 draws ########

 ###### marginal for theta1 ########
hist(theta1[500:5000])
summary(theta1[500:5000])
sd(theta1[theta1[500:5000])

######## marginal for theta2 ########
hist(theta2[500:5000])
summary(theta2[500:5000])
sd(theta2[500:5000])


####### plot the traces of the markov process #####
plot(theta1[500:5000], type="l", lty=1)
lines(theta2[500,5000],type="l",lty=2)

######## joint posterior for theta1 theta2 ######

plot(theta1[500:5000],theta2[500:5000],type="l")    # line plot to see the traces
plot(theta1[500:5000],theta2[500,5000], type="pt")  # point plot



#####################################
```

8.3 Exercise

Start the iterations at $\theta 2 = -2.5$ and repeat what we did above. Turn in the plots and the summary statistics. Put some titles on the plots.

8.4 Example

Remember the airline fatal accidensts ? I will give in class how to set it up as a hierarchical model and you will do it in the lab on Friday.