ST440/540 Applied Bayesian Analysis
Lab activity for 2/26/2024

Announcements

There is a group formation survey due next Friday on Moodle

For 540 students, I posted the project description the course assignments page.
https://st540.wordpress.ncsu.edu/assignments/
The next step is an abstract with a brief project description due in a few weeks.

For 440 students, your final exam will be a group analysis of a problem I assign you. Please still fill out the group formation survey! However, you do not need to start thinking of a research topic because I will send the project after the second mid-term.

A. HOMEWORK AND CLASS PARTICIPATION SOLUTIONS

None this week

B. DISCUSSION QUESTIONS

(1) Recall that Bayes' rule is $p(\theta|Y) = f(Y|\theta)\pi(\theta)/m(Y)$. Explain why we never need to compute m(Y) to perform Metropolis sampling. Your answer must include a formula!

Say θ_1 is the candidate and θ_0 is the previous value. The metropolis ratio is

$$R = \frac{p(\theta_1|Y)}{p(\theta_0|Y)} = \frac{f(Y|\theta_1)\pi(\theta_1)/m(Y)}{f(Y|\theta_0)\pi(\theta_0)/m(Y)} = \frac{f(Y|\theta_1)\pi(\theta_1)}{f(Y|\theta_0)\pi(\theta_0)}$$

and the constant m(Y) cancels so we never need to compute it.

(2) Assume the model $Y|\theta \sim \text{Gamma}(\theta,1)$ and prior $\theta \sim \text{Uniform}(0,10)$. This is not a conjugate prior and so you will use Metropolis-Hastings sampling.

(a) What is a reasonable candidate distribution for θ?

$\text{Normal}(\theta_0, c^2)$ is fine, although other distributions with support (0,10) might be more efficient.

(b) Give a formula for the acceptance probability (preferably in R code)

$$R = \frac{dgamma(Y,\theta_1,1) * dunif(\theta_1,0,10)}{dgamma(Y,\theta_0,1) * dunif(\theta_0,0,10)}$$

(c) What would you do if a candidate was outside the prior range (0,10)?

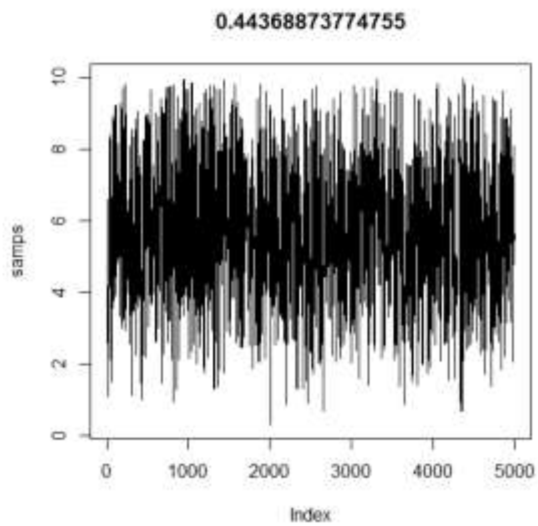
The prior PDF $dunif(\theta_1,0,10)$ is zero and so the candidate is automatically rejected.

(d) How would you tune the candidate distribution? Be specific.

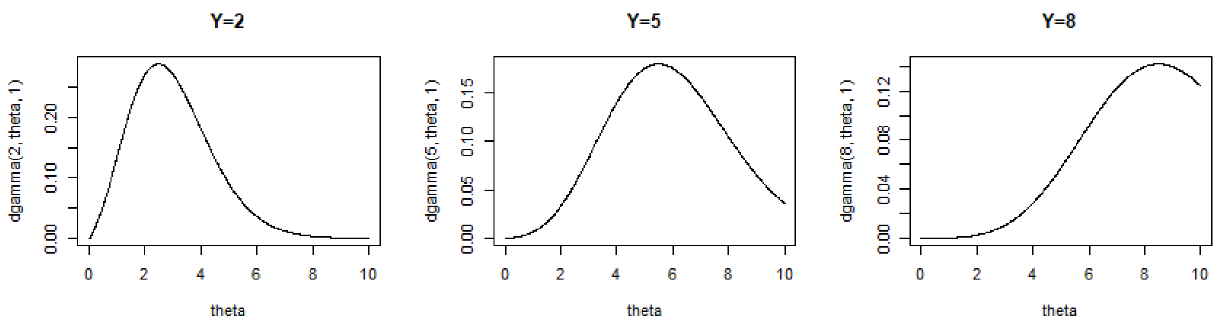
I would pick c until the acceptance probability is around 0.4.

Here is code if you're interested

```
Y      <- 5    # Data (I just picked something to illustrate the code)
can_sd <- 5    # Tuning parameter
iters  <- 5000 # Number of iters
theta  <- 1    # Initial value
samps  <- rep(theta,iters)
for(iter in 1:iters){
  can <- rnorm(1,theta,can_sd)
  if(can>0 & can<10){
     R   <- (dgamma(Y,can,1)*dunif(can,0,10))/
            (dgamma(Y,theta,1)*dunif(theta,0,10))
     if(runif(1)<R){theta <- can}
  }
  samps[iter] <- theta
}
acc_prob <- mean(samps[2:iters]!=samps[2:iters -1])
plot(samps,type="l",main=acc_prob)
```



These plots might help you visualize the problem:



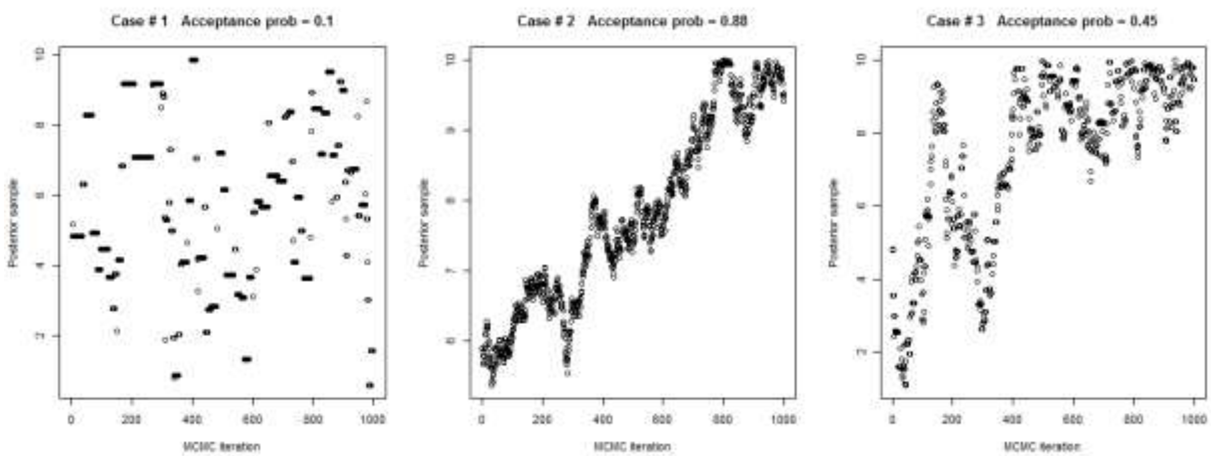

```
par(mfrow=c(1,3))
theta <- seq(0,10,.01)
plot(theta,dgamma(2,theta,1),type="l",main="Y=2")
plot(theta,dgamma(5,theta,1),type="l",main="Y=5")
plot(theta,dgamma(8,theta,1),type="l",main="Y=8")
```

(3) Referring to the model in (2), assume that we used a Gaussian candidate distribution with mean set to the previous value of θ and standard deviation c. The chains are run for 1000 iterations. For each of these plots, how would you modify the value c?

Case #1 – Lower c because acceptance is too low (maybe try c = c*0.8)

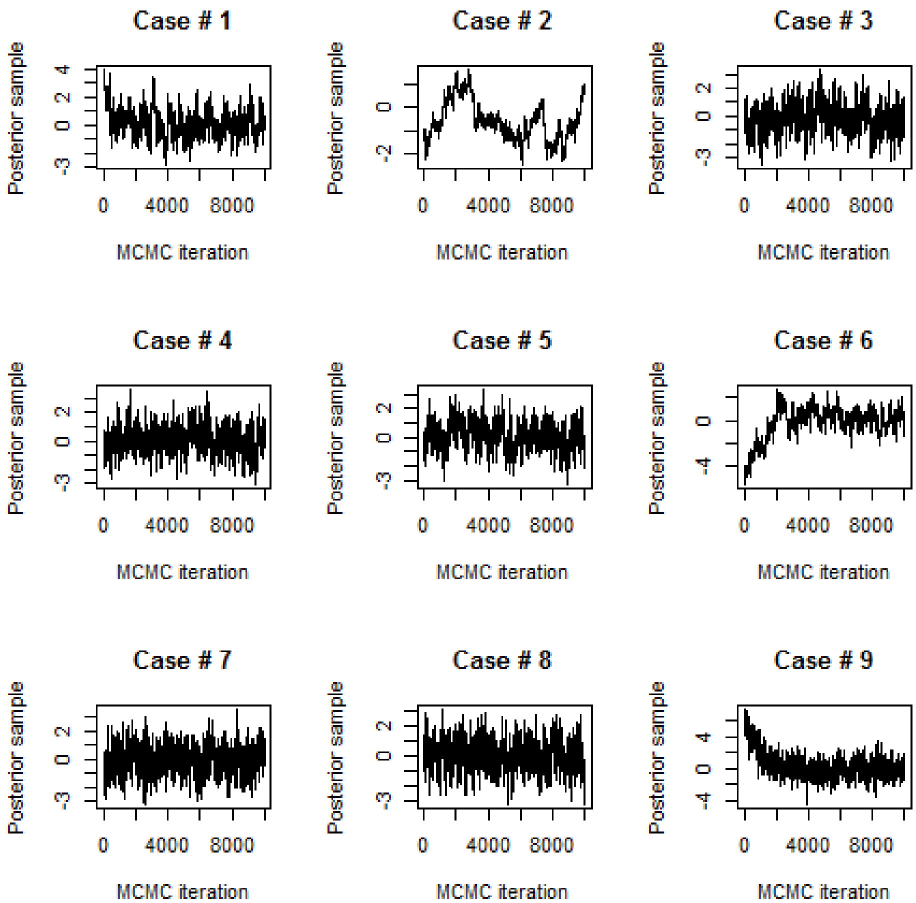
Case #2 – Increase c because acceptance is too high (maybe try c = c*1.2)

Case #3 – Looks pretty good, run it longer



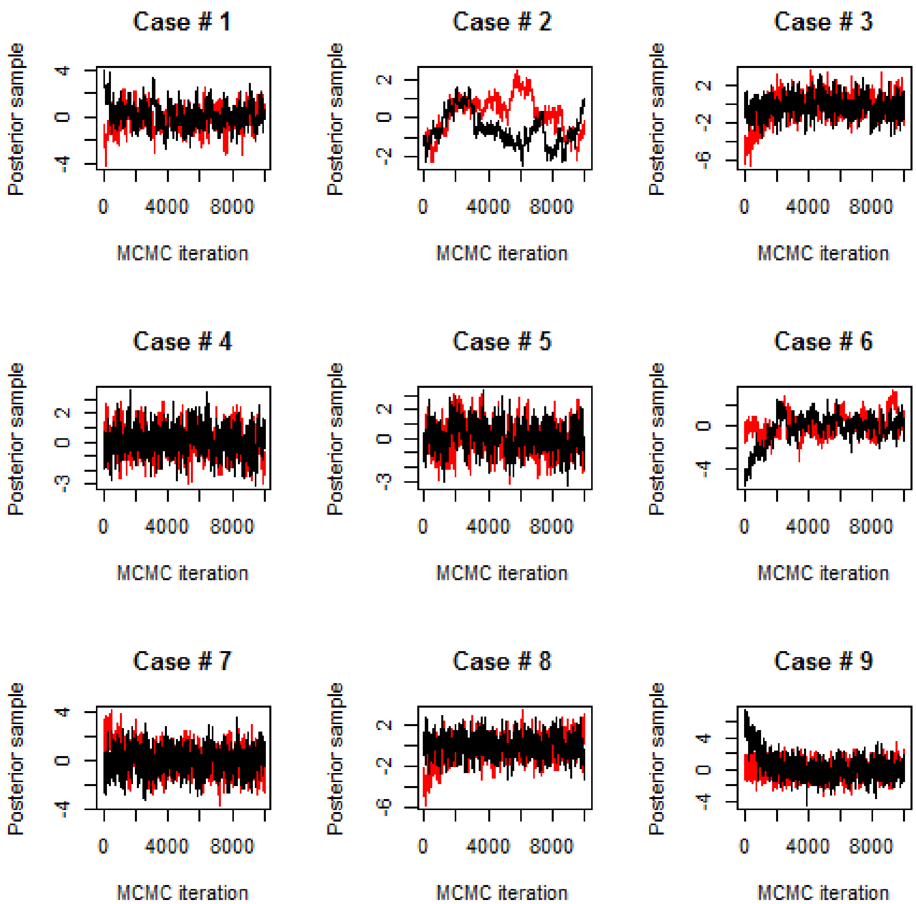
(4) For each trace plot below, at which iteration would you say the chain has converged?

Case #1 - 1000 Case #2 -10K Case #3 - 5K? Case #4 - 1K? Case #5 – 1K?
Case #6 -5K Case #7 - 1 Case #8 - 1 Case #9 – 5K



(5) Now instead of a single chain, two separate chains (one in red, one in black) are run with different starting values. For each case, select an iteration number T so that all samples after T from both chains can be kept and comment on how using multiple chains helped in this decision.

Case #1 - 1000 Case #2 -10K+ Case #3 - 5K Case #4 - 1 Case #5 – 1
Case #6 -5K Case #7 - 1 Case #8 - 1K Case #9 – 1K



(6) Consider the model Y|θ,b ~ Binomial(n,θ), θ|b ~ Beta(b,1-b) and b ~ Uniform(0,1).

(a) Specify initial values of θ and b.

See step 0 below

(b) What is the full conditional distribution of θ?

See step 1 below

(c) The full conditional distribution of b does not have a nice form and therefore can't be updated using Gibbs sampling. Sketch a Metropolis-within-Gibbs sampler for the joint posterior of (θ,b).

(0) Set theta = ½ and b = ½ (or theta = b= Y/n)

(1) Update theta given b as theta|b,Y ~ Beta(Y+b,n-Y+1-b)

(2) Update b given theta as

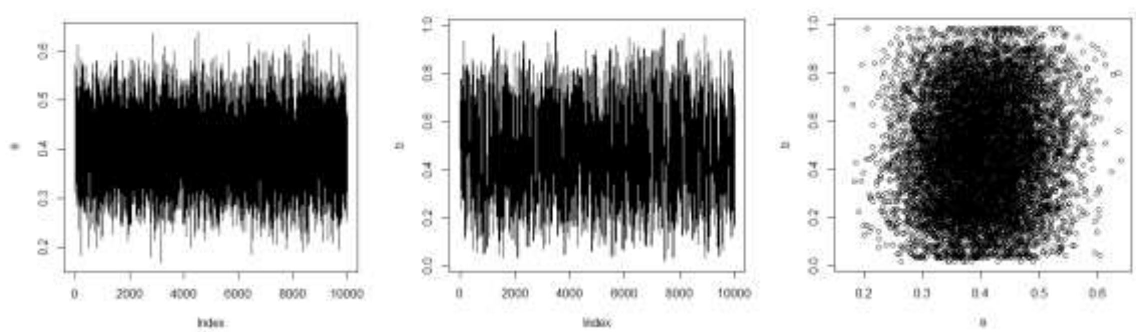
- Propose b_new |b_old ~ beta with mean b_old

Repeat steps (1) and (2) S times.

Here are code and results

```
n      <- 50
Y      <- 20
iters  <- 10000
tuning <- 1       # MH tuning parameter (see plots at the end)

theta  <- 0.5 # Initial values
b      <- 0.5

samps            <- matrix(0,iters,2)
samps[1,]        <- c(theta,b)
colnames(samps) <- c("theta","b")

for(iter in 1:iters){

  # Gibbs for theta
  theta <- rbeta(1,Y+b,n-Y+1-b)

  # MH for b
  can <- rbeta(1,tuning*b,tuning*(1-b))
  R1  <- dbeta(theta,can,1-can)*
         dunif(can,0,1)*
         dbeta(b,tuning*can,tuning*(1-can))
  R2  <- dbeta(theta,b,1-b)*
         dunif(b,0,1)*
         dbeta(can,tuning*b,tuning*(1-b))
  R   <- R1/R2
  if(runif(1)<R){b<-can}

  samps[iter,] <- c(theta,b)
}
```

```
acc_prob <- colMeans(samps[2:iters,]!=samps[2:iters -1, ])
> acc_prob
    theta         b
1.0000000 0.3163316
>
> colMeans(samps)
    theta         b
0.4004075 0.4765602
>
> apply(samps,2,quantile,c(0.025,0.975))
          theta          b
2.5%  0.2698998 0.07935664
97.5% 0.5384169 0.89343295

plot(samps[,1],ylab=expression(theta),type="l")
plot(samps[,2],ylab=expression(b),type="l")
plot(samps,xlab=expression(theta),ylab=expression(b))
```



```
# Plots of the candidate distribution
b <- seq(0,1,.01)
tuning <- 10; old <- .3
plot(b,dbeta(b,tuning*old,tuning*(1-old)),type="l")
tuning <- 10; old <- .7
plot(b,dbeta(b,tuning*old,tuning*(1-old)),type="l")
tuning <- 100; old <- .7
plot(b,dbeta(b,tuning*old,tuning*(1-old)),type="l")
```