Statistical Computing Homework 7, Chapter 6

Ziqi Yang

26 October, 2018

Contents

R Markdown

##Generate the sample

```r
set.seed(20181026)

library(HI)
# Set the underlying distribution
mu1 <- 3; sigma1 <- 5;   mu2 <- 5;  sigma2 <- 5
p1 <- 0.7;  p2 <- 0.3
n <- 5000

s.sample <- numeric(n)
for (i in 1:n) {
  u <- runif(1)
  if (u < p1) {
    s.sample[i] <- rnorm(1, mean = mu1, sd = sigma1)
  } else {
    s.sample[i] <- rnorm(1, mean = mu2, sd = sigma2)
  }
}
```

##Set the posterior distribution

```r
log.post <- function(x1, x2, x3, x4) {
  x<- c(x1, x2, x3, x4)
  sum( log( p1 * dnorm(s.sample, mean = x[1], sd = x[2]) + p2 * dnorm(s.sample, mean = x[3], sd = x[4])
    log(dnorm(x[1], mean = 0, sd = 10)) +
    log(dgamma(x[2]^(-2), shape = 0.5, scale = 10)) +
    log(dnorm(x[3], mean = 0, sd = 10)) +
    log(dgamma(x[4]^(-2), shape = 0.5, scale = 10))
}
```

##Initial values for arms

```r
n.post.sample <- 5000*4
init <- numeric(n.post.sample)
init <- runif(n.post.sample, min = rep(c(-10,-0), n.post.sample/2), max = 10)
```

##Use the Gibbs Sampling

```r
n.post.sample <- 5000
post.sample <- matrix(0, nrow = n.post.sample, ncol = 4)
post.sample[1, ] <- c(0,1,0,1)
k <- 1
for (j in 1:(n.post.sample-1)) {
  post.sample[j+1, 1] <- arms( init[k], function(x) log.post( x, post.sample[j,2], post.sample[j,3], pos
  k <- k + 1
  post.sample[j+1, 2] <- arms( init[k], function(x) log.post( post.sample[j+1,1], x, post.sample[j,3], p
  k <- k + 1
  post.sample[j+1, 3] <- arms( init[k], function(x) log.post( post.sample[j+1,1], post.sample[j+1,2], x
  k <- k + 1
  post.sample[j+1, 4] <- arms( init[k], function(x) log.post( post.sample[j+1,1], post.sample[j+1,2], po
  k <- k + 1
}
```

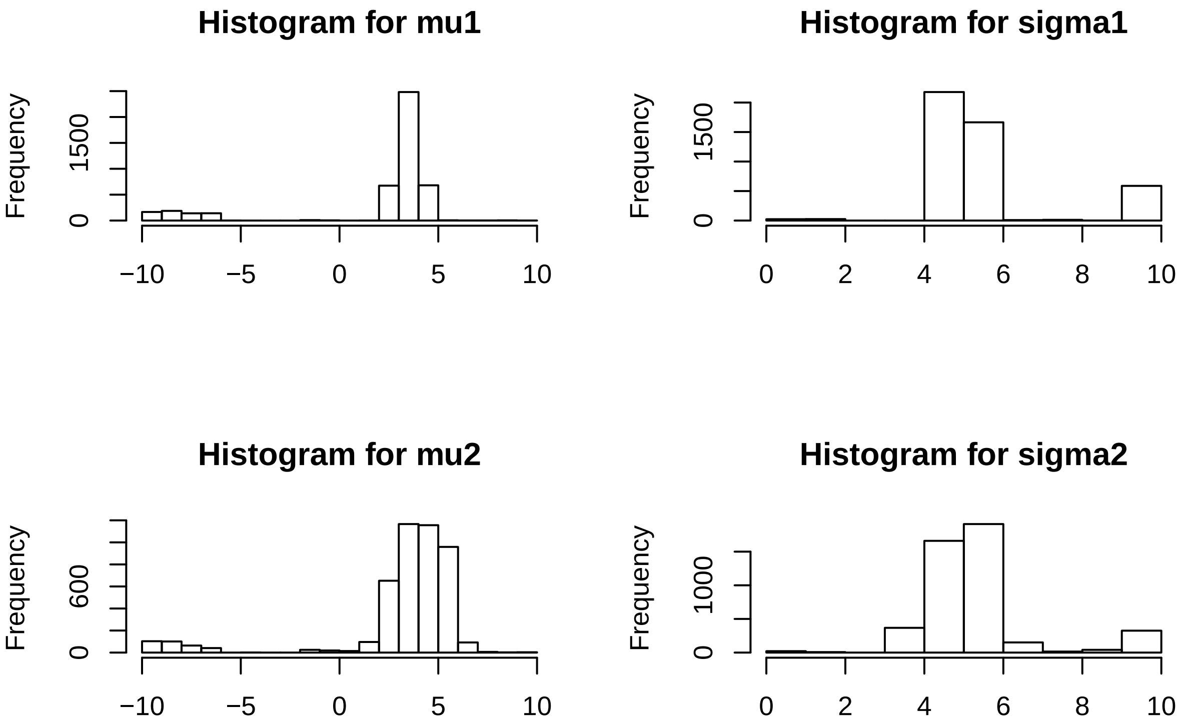
In each Gibbs sampling update, we use the Adaptive Rejection Metropolis Sampling from HI package. And actually here we can prove that the marginals posterior density for each paramters is log-concave, as long as we have sample size greater than 1

##Draw the graph

```
burn <- 500
colMeans(post.sample[(burn+1):n.post.sample, ])
```

```
## [1] 1.847295 5.571675 3.181404 5.349567
```

```
par(mfrow=c(2,2))
hist(post.sample[(burn+1):n.post.sample,1], main = "Histogram for mu1", xlab = NULL)
hist(post.sample[(burn+1):n.post.sample,2], main = "Histogram for sigma1", xlab= NULL)
hist(post.sample[(burn+1):n.post.sample,3], main = "Histogram for mu2", xlab = NULL)
hist(post.sample[(burn+1):n.post.sample,4], main = "Histogram for sigma2", xlab = NULL)
```



```
par(mfrow=c(1,1))
```

The initial values for the parameters are 3, 5, 5, 5 for $\mu_1, \sigma_1, \mu_2, \sigma_2$ respectively. We can see our

final posterior histograms match the result