

MCMC by Gibbs Sampling

HW 7 of STAT 5361 Statistical Computing

*Biju Wang**

10/23/2018

1 MCMC by Gibbs Sampling

The distribution of X is finite mixture normal distribution as follows

$$f(x) = \delta N(\mu_1, \sigma_1^2) + (1 - \delta)N(\mu_2, \sigma_2^2)$$

with $\delta = 0.7, \mu_1 = 7, \sigma_1^2 = 0.5^2, \mu_2 = 10, \sigma_2^2 = 0.5^2$.

And we sample $n = 100$ from this mixture distribution and denoted as $\mathbf{x} = (x_1, \dots, x_{100})'$. The likelihood function is

$$L(\mathbf{x}; \delta, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2) = \prod_{i=1}^n \left[\delta \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_i - \mu_1)^2}{2\sigma_1^2}} + (1 - \delta) \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_i - \mu_2)^2}{2\sigma_2^2}} \right]$$

We impose an uninformative prior on δ , the prior for μ_1 and μ_2 are $N(0, 10^2)$, the prior for $\frac{1}{\sigma_1^2}$ and $\frac{1}{\sigma_2^2}$ are $\Gamma(0.5, 10)$, hence the prior for σ_1^2 and σ_2^2 are $IG(0.5, 10)$ which is inverse gamma distribution. The density for inverse gamma is

$$g(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} \left(\frac{1}{x}\right)^{\alpha+1} e^{-\frac{1}{x\beta}} \quad x > 0$$

where α is the shape parameter and β is the scale parameter.

Thus the posterior density is

$$\begin{aligned} \pi(\delta, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2 | \mathbf{x}) &\propto \prod_{i=1}^n \left[\delta \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_i - \mu_1)^2}{2\sigma_1^2}} + (1 - \delta) \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_i - \mu_2)^2}{2\sigma_2^2}} \right] \times \frac{1}{\sqrt{2\pi} \times 10} e^{-\frac{\mu_1^2}{2 \times 10^2}} \times \\ &\quad \frac{1}{\sqrt{2\pi} \times 10} e^{-\frac{\mu_2^2}{2 \times 10^2}} \times \frac{1}{\sqrt{\pi} \times 10^{\frac{1}{2}}} \left(\frac{1}{\sigma_1^2}\right)^{\frac{3}{2}} e^{-\frac{1}{10 \times \sigma_1^2}} \times \frac{1}{\sqrt{\pi} \times 10^{\frac{1}{2}}} \left(\frac{1}{\sigma_2^2}\right)^{\frac{3}{2}} e^{-\frac{1}{10 \times \sigma_2^2}} \end{aligned}$$

We now need to construct a markov chain by Gibbs sampling such that the stationary of the markov chain is this posterior distribution. In order to implement Gibbs sampling, we need to be able to sample from the following

$$\begin{aligned} &\pi(\delta | \mathbf{x}, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2) \quad \pi(\mu_1 | \mathbf{x}, \delta, \sigma_1^2, \mu_2, \sigma_2^2) \\ &\pi(\sigma_1^2 | \mathbf{x}, \delta, \mu_1, \mu_2, \sigma_2^2) \quad \pi(\mu_2 | \mathbf{x}, \delta, \mu_1, \sigma_1^2, \sigma_2^2) \quad \pi(\sigma_2^2 | \mathbf{x}, \delta, \mu_1, \sigma_1^2, \mu_2) \end{aligned}$$

Rejection sampling can do this, adaptive rejection sampling (ARS) can efficiently sample from univariate log-concave distributions. The Metropolis-Hastings (MH) algorithm can also be used to sample from each conditional distribution. We can use MH algorithm, but in order to avoid high probabilities of rejection (and hence slower convergence of the chain), it may be helpful to adapt the proposal density $k(x|y)$ to the shape of the full conditional density $\pi(\cdot|\cdot)$. Since ARS provides a way of adapting a function to $\pi(\cdot|\cdot)^{[1]}$, thus ARS and MH algorithm can be combined to improve the efficiency of MH algorithm. This is the so-called adaptive rejection metropolis sampling (ARMS). In this problem, we will use ARMS to sample from each conditional density.

The initial values are $\delta^{(0)} = 0.5, \mu_1^{(0)} = 1, \mu_2^{(0)} = 1, \sigma_1^{2(0)} = 1, \sigma_2^{2(0)} = 1$. We run 2500 iterations and throw away the first 500 iterations.

*bijuwang@uconn.edu

```

delta <- 0.7
n <- 100
set.seed(254)
u <- rbinom(n, prob = delta, size = 1)
x <- rnorm(n, ifelse(u == 1, 7, 10), 0.5)
x1 <- x

#define log-likelihood function
log.likelihood <- function(delta, mu1, mu2, sigma12, sigma22, x = x1){
  sum(log(delta * dnorm(x, mu1, sqrt(sigma12)) + (1 - delta) * dnorm(x, mu2, sqrt(sigma22))))
}

library("invgamma")
#define log prior function
log.prior <- function(delta, mu1, mu2, sigma12, sigma22){
  delta.logprior <- 0
  mu1.logprior <- dnorm(mu1, 0, 10, log = T)
  mu2.logprior <- dnorm(mu2, 0, 10, log = T)
  sigma12.logprior <- dinvgamma(sigma12, shape = 0.5, scale = 10, log = T)
  sigma22.logprior <- dinvgamma(sigma22, shape = 0.5, scale = 10, log = T)

  sum(delta.logprior + mu1.logprior + mu2.logprior +
      sigma12.logprior + sigma22.logprior)
}

#define log posterior
log.posterior <- function(delta, mu1, mu2, sigma12, sigma22, x = x1){
  log.likelihood(delta, mu1, mu2, sigma12, sigma22, x) +
  log.prior(delta, mu1, mu2, sigma12, sigma22)
}

library("HI")
#Gibbs Sampling
mygibbs <- function(delta.init, mu1.init, mu2.init, sigma12.init, sigma22.init, x = x1, niter){
  init <- c(delta.init, mu1.init, mu2.init, sigma12.init, sigma22.init)
  chain <- matrix(NA, nr = niter, nc = length(init))

  for (i in 1:niter) {
    chain[i, 1] <- arms(runif(1, 0, 1), log.posterior, function(x, ...) (x > 0) * (x < 1), 1,
                      mu1 = init[2], mu2 = init[3], sigma12 = init[4], sigma22 = init[5])
    init[1] <- chain[i, 1]
    chain[i, 2] <- arms(0, log.posterior, function(x, ...) (x > -100) * (x < 100), 1,
                      delta = init[1], mu2 = init[3], sigma12 = init[4], sigma22 = init[5])
    init[2] <- chain[i, 2]
    chain[i, 3] <- arms(0, log.posterior, function(x, ...) (x > -100) * (x < 100), 1,
                      delta = init[1], mu1 = init[2], sigma12 = init[4], sigma22 = init[5])
    init[3] <- chain[i, 3]
    chain[i, 4] <- arms(runif(1, 1e-4, 200), log.posterior, function(x, ...) (x > 1e-4) * (x < 200), 1,
                      delta = init[1], mu1 = init[2], mu2 = init[3], sigma22 = init[5])
    init[4] <- chain[i, 4]
    chain[i, 5] <- arms(runif(1, 1e-4, 200), log.posterior, function(x, ...) (x > 1e-4) * (x < 200), 1,
                      delta = init[1], mu1 = init[2], mu2 = init[3], sigma12 = init[4])
    init[5] <- chain[i, 5]
  }
}

```

```

}

chain
}

niter <- 2500
results <- mygibbs(0.5, 1, 1, 1, 1, niter = niter)[- (1:500),]

library("ggplot2")
library("gridExtra")
p1 <- ggplot(data.frame(x = results[,1]), aes(x = x)) +
  geom_histogram(aes(y=..density..), color = "black", alpha = 0.2) +
  labs(x = expression("Values of"~delta), y = expression("Marginal Density of"~delta)) +
  theme(plot.title = element_text(hjust = 0.5)) + ggtitle(expression("Histogram for"~delta))

p1

```

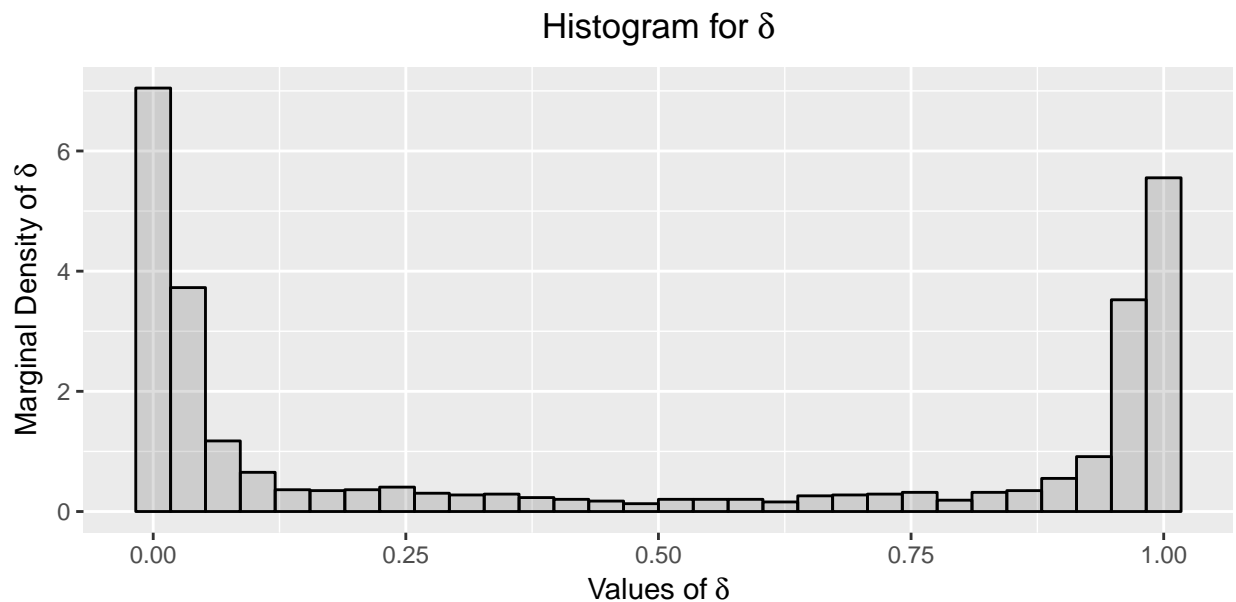


Figure 1: Histogram for δ

```

p2 <- ggplot(data.frame(x = results[,2]), aes(x = x)) +
  geom_histogram(aes(y=..density..), color = "black", alpha = 0.2) +
  labs(x = expression("Values of"~mu[1]), y = expression("Marginal Density of"~mu[1])) +
  theme(plot.title = element_text(hjust = 0.5)) + ggtitle(expression("Histogram for"~mu[1]))

p3 <- ggplot(data.frame(x = results[,3]), aes(x = x)) +
  geom_histogram(aes(y=..density..), color = "black", alpha = 0.2) +
  labs(x = expression("Values of"~mu[2]), y = expression("Marginal Density of"~mu[2])) +
  theme(plot.title = element_text(hjust = 0.5)) + ggtitle(expression("Histogram for"~mu[2]))

grid.arrange(p2, p3, nrow = 2)

```

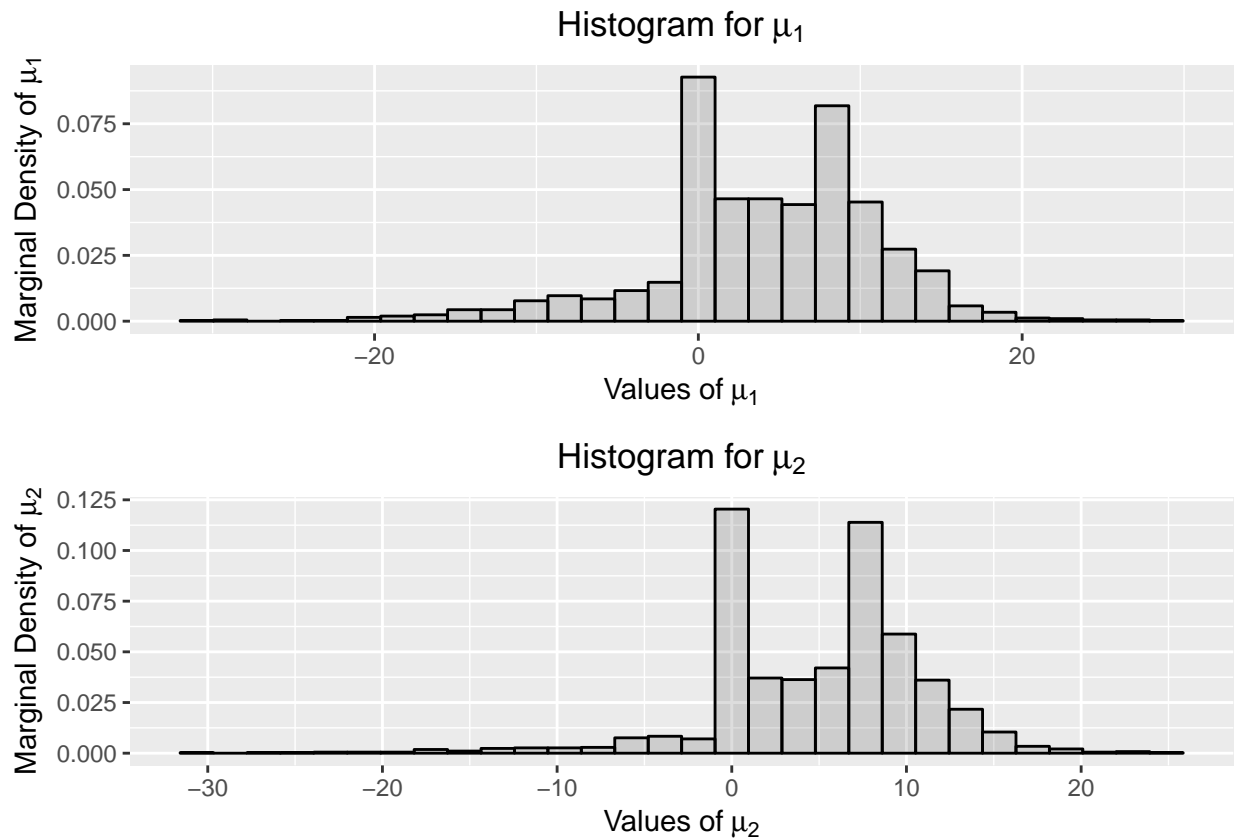


Figure 2: Histograms for μ_1, μ_2

```
p4 <- ggplot(data.frame(x = results[,4]), aes(x = x)) +
  geom_histogram(aes(y=..density..), color = "black", alpha = 0.2) +
  labs(x = expression("Values of"~sigma[1]^2), y = expression("Marginal Density of"~sigma[1]^2)) +
  theme(plot.title = element_text(hjust = 0.5)) + ggtitle(expression("Histogram for"~sigma[1]^2))

p5 <- ggplot(data.frame(x = results[,5]), aes(x = x)) +
  geom_histogram(aes(y=..density..), color = "black", alpha = 0.2) +
  labs(x = expression("Values of"~sigma[2]^2), y = expression("Marginal Density of"~sigma[2]^2)) +
  theme(plot.title = element_text(hjust = 0.5)) + ggtitle(expression("Histogram for"~sigma[2]^2))

grid.arrange(p4, p5, nrow = 2)
```

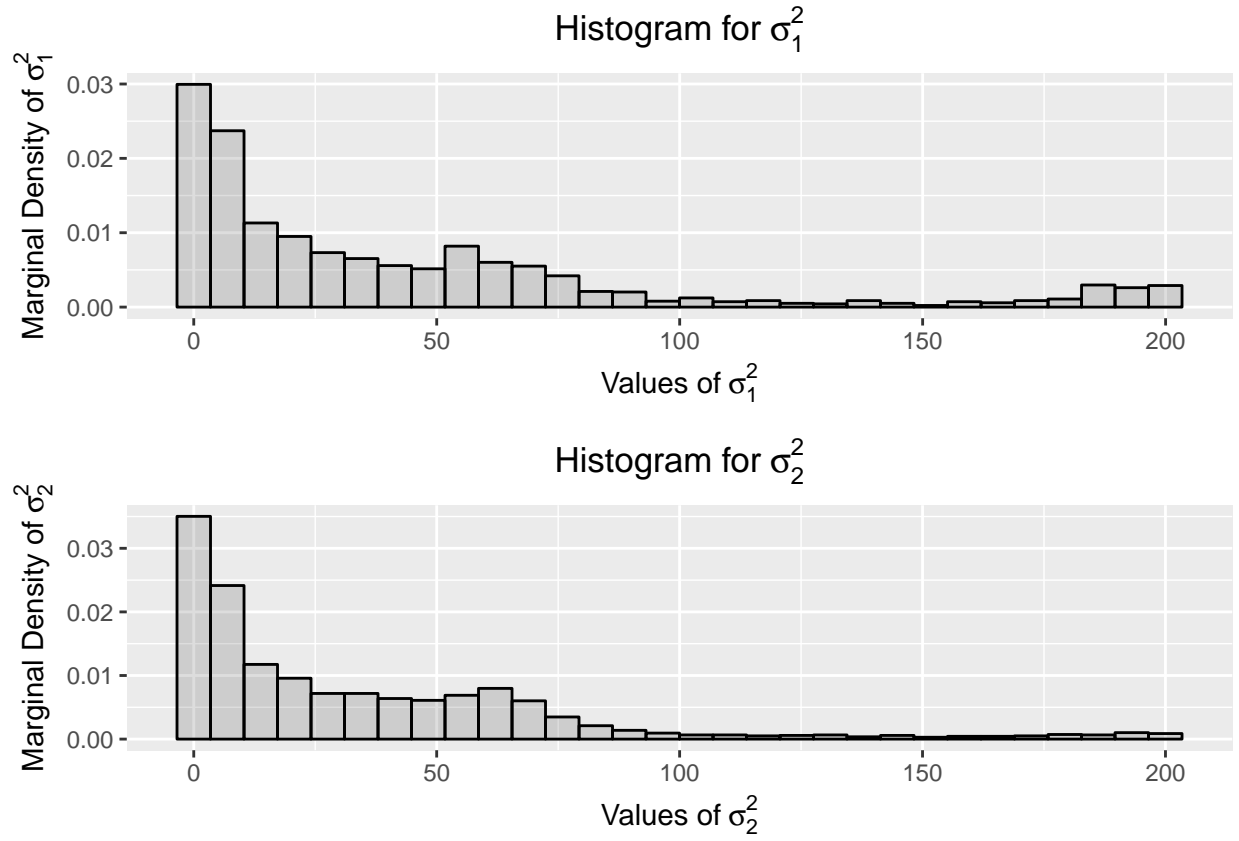


Figure 3: Histograms for σ_1^2, σ_2^2

2 References

1. Gilks, W. R., N. G. Best, and K. K. C. Tan. "Adaptive Rejection Metropolis Sampling within Gibbs Sampling." *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 44, no. 4 (1995): 455-472.