Statistical Computing Homework 7

Jieying Jiao

26 October 2018

Abstract

This is Jieying Jiao's homework 7 for statistical computing, fall 2018.

Contents

1 Exercise 6.3.1

1.1 posterior distribution

From the definition, we can easily write down the posterior distribution:

$$p(\theta|\mathbf{x}) \propto \prod_{i=1}^{n}[\frac{\delta}{\sigma_1}\exp\{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}\} + \frac{1-\delta}{\sigma_2}\exp\{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}\}]\times$$
$$\exp\{-\frac{\mu_1^2}{200} - \frac{\mu_2^2}{200} - \frac{1}{10\sigma_1^2} - \frac{1}{10\sigma_2^2}\}\sigma_1^{-2}\sigma_2^{-2}$$

Then:

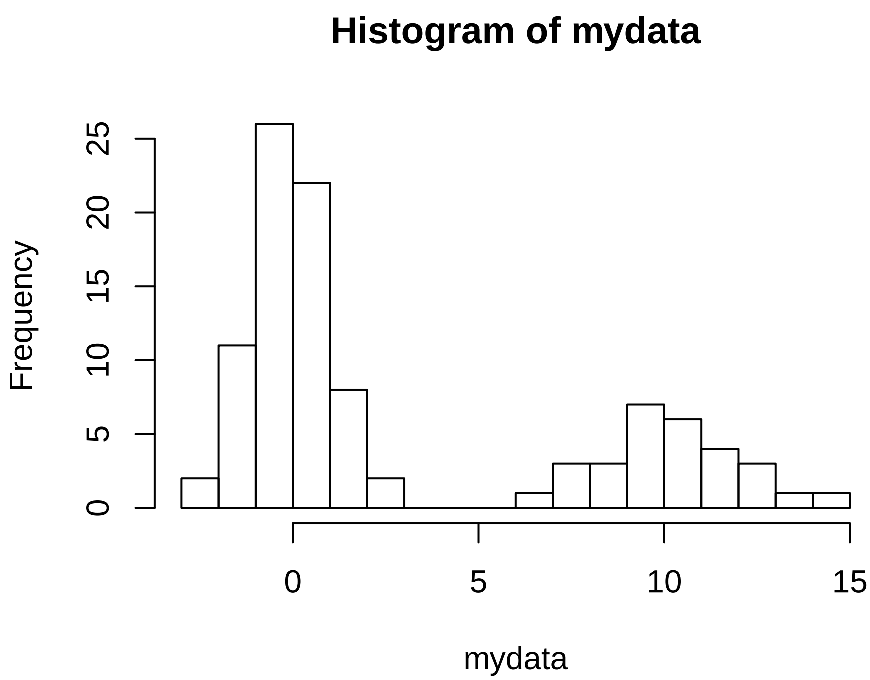
$$\log p(\theta|\mathbf{x}) = C + \sum_{i=1}^{n}\log[\frac{\delta}{\sigma_1}\exp\{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}\} + \frac{1-\delta}{\sigma_2}\exp\{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}\}] -$$
$$\frac{\mu_1^2}{200} - \frac{\mu_2^2}{200} - \frac{1}{10\sigma_1^2} - \frac{1}{10\sigma_2^2} - 2\log\sigma_1^2 - 2\log\sigma_2$$

where $\theta = (\mu_1, \mu_2, \sigma_1, \sigma_2, \delta)$, $\mathbf{x} = (x_1, \ldots, x_n)$ are data we have, C is the normalization constant.

```r
library(HI)
```

1.2 data generation

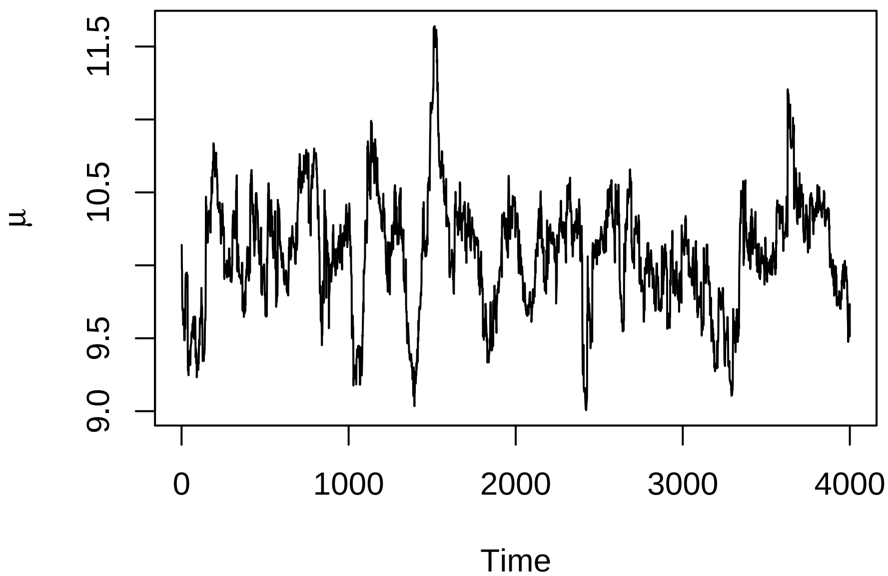
```r
## Generate data with size n = 100
## mixture proportion delta = 0.7, mu1 = 0, sigma1 = 1, mu2 = 10, sigma2 = 2
n <- 100
delta <- 0.7
mu1 <- 0
mu2 <- 10
sigma1 <- 1
sigma2 <- 2
set.seed(123)
u <- rbinom(n, size = 1, prob = delta)
mydata <- rnorm(n, ifelse(u == 1, mu1, mu2), ifelse(u == 1, sigma1, sigma2))
hist(mydata, nclass = 20)
```



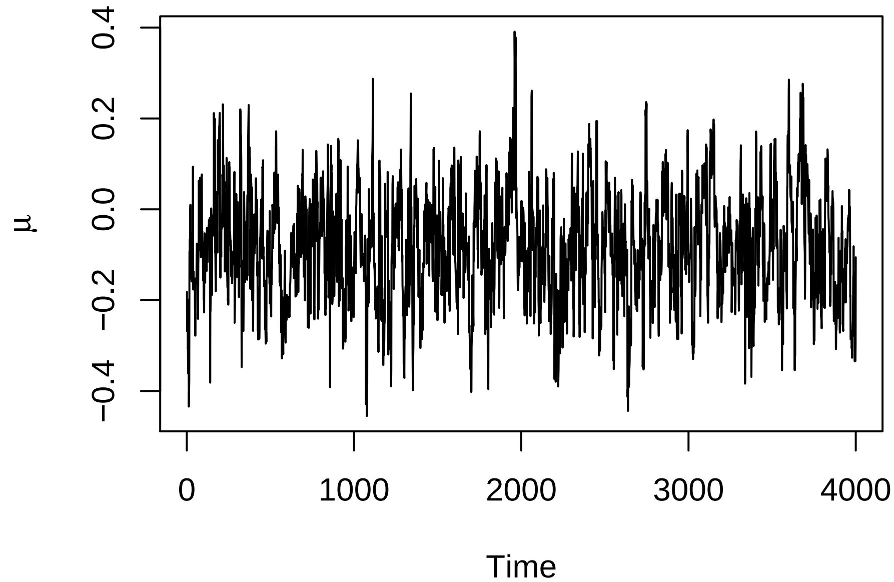

```r
## define the log-posterior density function without the normalization constant
logpost <- function(theta) {
  mu1 <- theta[1]
  mu2 <- theta[2]
  sigma1 <- theta[3]
  sigma2 <- theta[4]
  delta <- theta[5]
  return(sum(log(delta * exp(-(mydata-mu1)^2/2/sigma1^2)/sigma1 + (1-delta) *
                  exp(-(mydata-mu2)^2/2/sigma2^2)/sigma2)) - mu1^2/200 -
           mu2^2/200 - 1/10/sigma1^2 - 1/10/sigma2^2 - 2*log(sigma1) - 2*log(sigma2))
}
```

1.3 Posterior distribution estimation using MCMC with Gibbs sampling

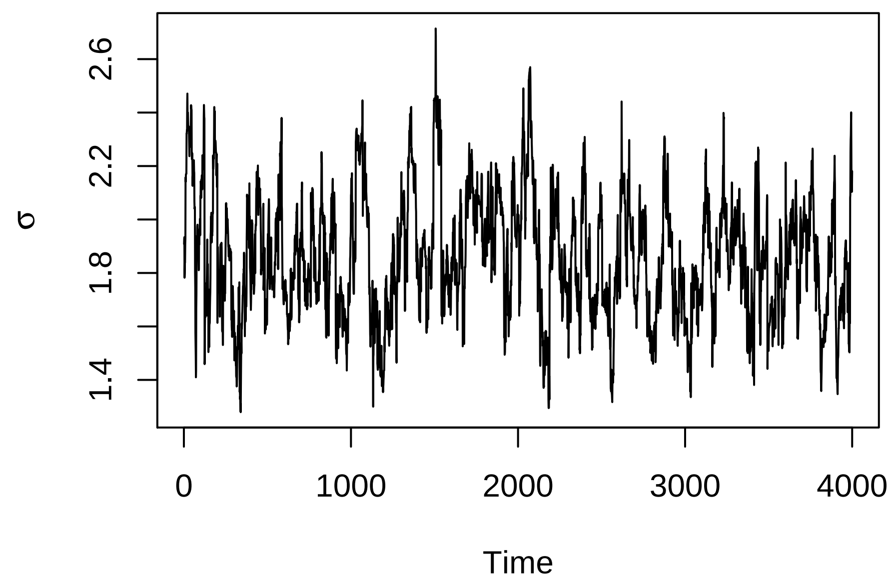
```r
## define support function
mysupp <- function(x) {
  x[1] <- 1 / (exp(x[1])+1)
  x[2] <- 1 / (exp(x[2])+1)
  x[3] <- 1 / (x[3]+1)
  x[4] <- 1 / (x[4] + 1)
  x[5] <- x[5]
  return((min(x)>0)*(max(x)<1))
}

## running MCMC using Gibbs sampling
y <- arms(c(0, 0, 1, 1, 0.2), logpost, mysupp, 10000)
y <- y[-(1:6000), ]
plot(ts(y[, 1]), ylab = expression(mu))
```



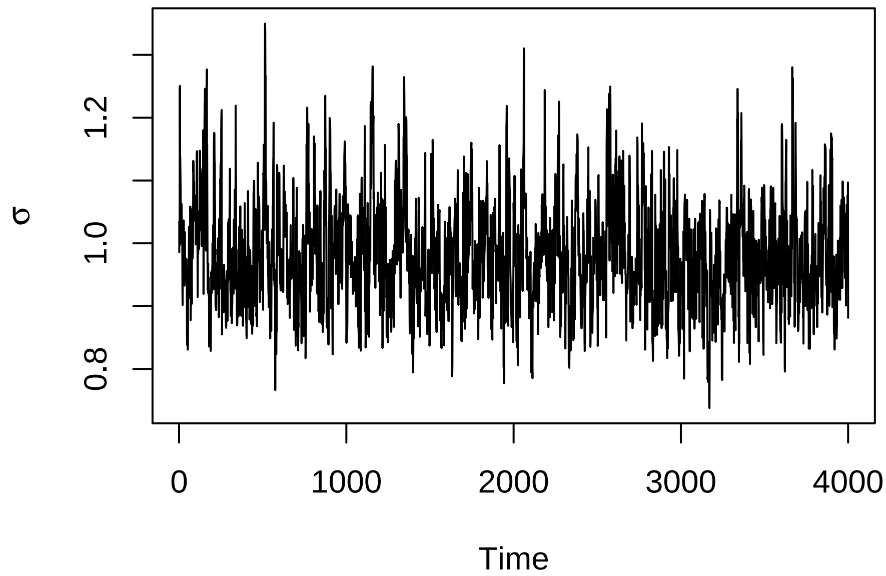

```r
plot(ts(y[, 2]), ylab = expression(mu))
```



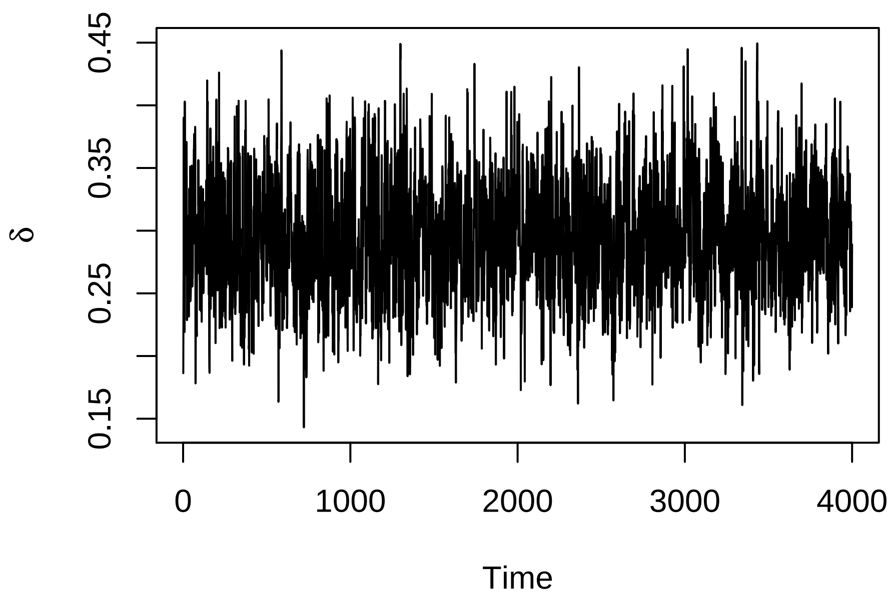

```
plot(ts(y[, 3]), ylab = expression(sigma))
```




```
plot(ts(y[, 4]), ylab = expression(sigma))
```




```r
plot(ts(y[, 5]), ylab = expression(delta))
```



The result we get switch the notation, but it doesn't matter, we still get the estimation that 0.7 proportion of population are from population $N(0, 1)$ and 0.3 proportion of population are from $N(10, 2)$.