

computational_statistic lab4

Group 8

11/21/2021

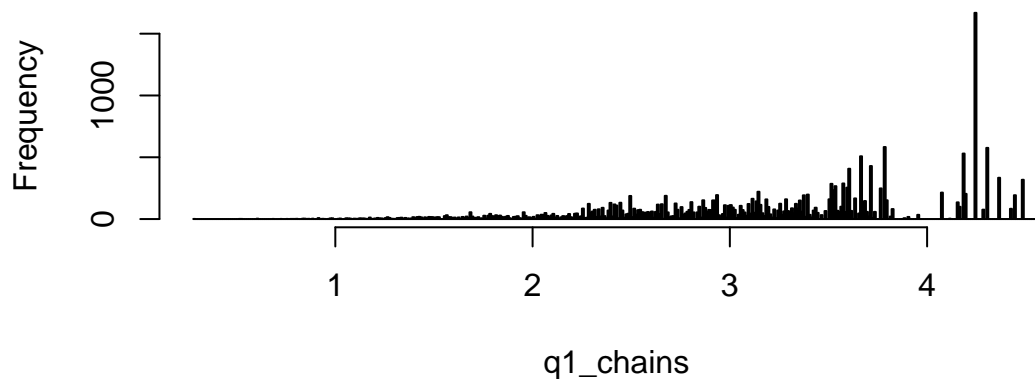
Question 1 Computations with Metropolis–Hastings

Q1: Use Metropolis–Hastings algorithm to generate samples from this distribution by using proposal distribution as log-normal $\text{LN}(X_t, 1)$, take some starting point. Plot the chain you obtained as a time series plot. What can you guess about the convergence of the chain?

Solution :

According to the Markov chains theory,I guess this plot will finally converge.

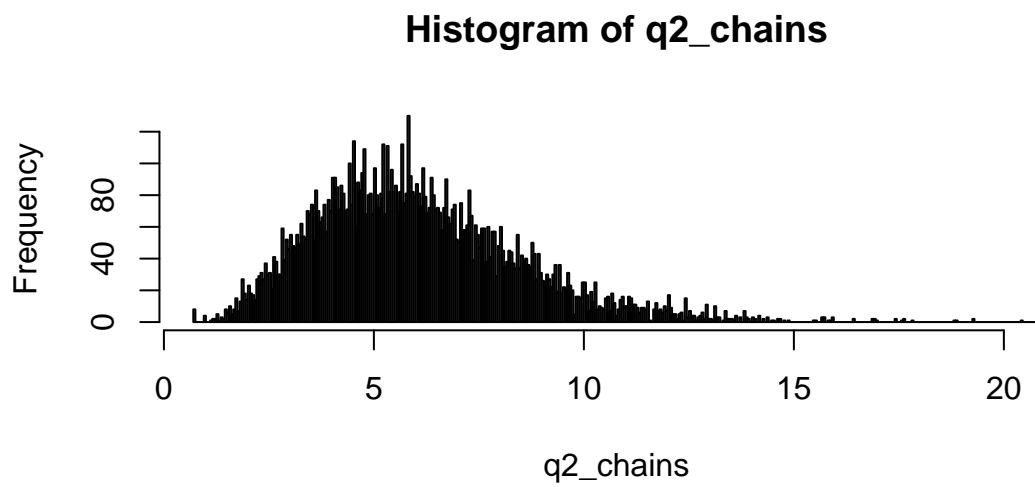
Histogram of q1_chains

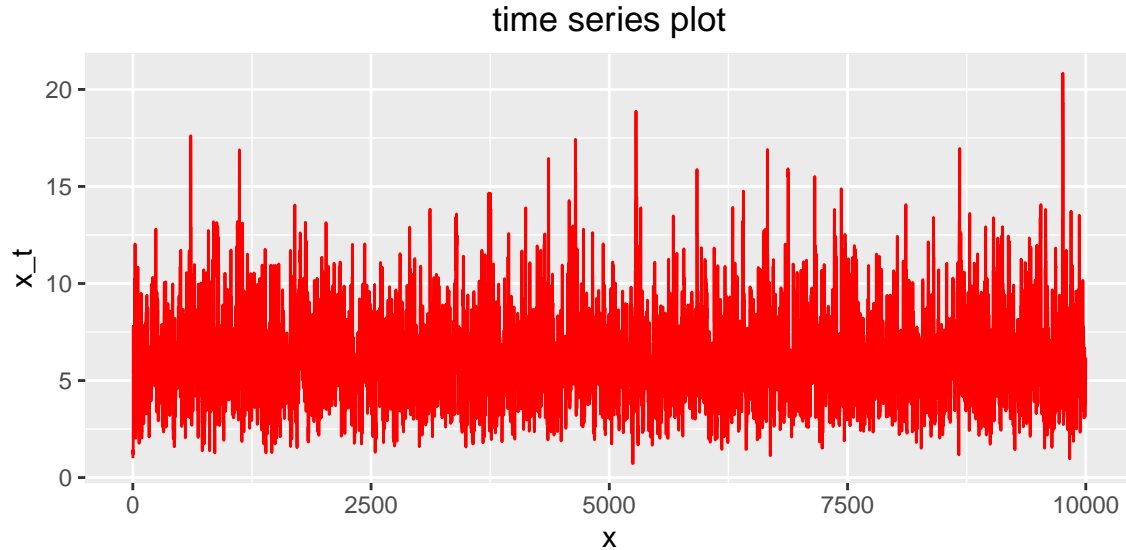




Q2: Perform Step 1 by using the chi-square distribution as a proposal distribution. What can you guess about the convergence of the chain?

Solution :





This chain is converge faster than what we have in the first question, and it's histogram is well matched the density function what we want to sample from.

Q3: Generate 10 MCMC sequences using the generator from Step 2 and starting points 1, 2, . . . , or 10. Use the Gelman–Rubin method to analyze convergence of these sequences.

Solution :

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1.01      1.01
```

the upper limit is less than 1.2.so the chains are converged

Q4: Estimate $\int_0^\infty xf(x)dx$

solution :

The sample from step 1 and 2 are stored in q1_chains and q2_chains. From the formula, We know that the integral is the estimation of the distribution(what we want to sample from). And now, we already have the samples from this distribution, so what we need to do is just compute the mean value of these samples, and we know that $\hat{\mu} = \text{mean}(\text{samples})$ is the unbiased estimator of μ where μ is the estimation.

```
## [1] "for sample from question1: 3.30614158635729"
```

```
## [1] "for sample from question1: 6.03734809789173"
```

As the result shows, chains from question 1 is not very good(we can tell from the histogram in question1),but samples from question 2 are much better. So, we can make a conclusion that a proporsal density function is very important in the Hasting sampling.

Q5: The distribution generated is in fact a gamma distribution (6, 1). Determine the actual value of the integral. Compare it with the one you obtained in the previous step.

solution:

The actual value of integral is estimation of gamma(6,1), which equals 6. From the results presented above, only the second samples can match.

Question 2 Gibbs sampling

Q1: Present the formula.

solution:

$$Y_i = \mathcal{N}(\mu, \sigma = 0.2), \quad i = 1, \dots, n$$

where the prior is

$$p(\mu_1) = 1$$

$$p(\mu_{i+1} | \mu_i) = \mathcal{N}(\mu_i, \sigma) \quad i = 1, \dots, n-1$$

$p(\vec{Y} | \vec{\mu})$ and $p(\vec{\mu})$ are:

$$p(\vec{Y} | \vec{\mu}, \sigma) = \prod_{i=1}^n \frac{1}{\sqrt{2\sigma\pi}} \exp\left(-\frac{(y_i - \mu_i)^2}{2\sigma}\right)$$

$$= \left(\frac{1}{\sqrt{2\sigma\pi}}\right)^n \exp\left(-\frac{\sum_{i=1}^n (y_i - \mu_i)^2}{2\sigma}\right)$$

$$p(\vec{\mu}) = p(\mu_1) \cdot p(\mu_2 | \mu_1) \cdot p(\mu_3 | \mu_2) \cdots p(\mu_n | \mu_{n-1})$$

$$= 1 \cdot \prod_{i=2}^n p(\mu_i | \mu_{i-1})$$

$$= \frac{1}{\sqrt{2\sigma\pi}} \exp\left(-\frac{(\mu_2 - \mu_1)^2}{2\sigma}\right) \cdots \exp\left(-\frac{(\mu_n - \mu_{n-1})^2}{2\sigma}\right)$$

$$= \left(\frac{1}{\sqrt{2\sigma\pi}}\right)^{n-1} \exp\left(-\frac{\sum_{i=2}^n (\mu_i - \mu_{i-1})^2}{2\sigma}\right)$$

Q2: Bayes's theorem.

solution:

$$p(\mu_1 | \mu_{-1}, \vec{Y}) = \frac{p(\vec{\mu}_{-1}, \mu_1, \vec{Y})}{\int p(\vec{Y}, \vec{\mu}) d\mu_1} = \frac{\prod_{i=2}^n p(Y_i | \mu_i) * \prod_{i=3}^n p(\mu_i | \mu_{i-1}) * p(\mu_2 | \mu_1) * p(Y_1 | \mu_1) * p(\mu_1)}{\prod_{i=2}^n p(Y_i | \mu_i) * \prod_{i=3}^n p(\mu_i | \mu_{i-1}) * \int p(\mu_2 | \mu_1) * p(Y_1 | \mu_1) * p(\mu_1) d\mu_1}$$

Since the integral part would be a constant, we can just discard it when we focus on the distribution. After we discard all constant part, we can get

$$p(\mu_1 | \vec{\mu}_{-1}, \vec{Y}) \propto p(\mu_2 | \mu_1) * p(Y_1 | \mu_1) * p(\mu_1)$$

$$\propto \exp\left(-\frac{(y_1 - \mu_1)^2 + (\mu_2 - \mu_1)^2}{2\sigma^2}\right)$$

$$\propto \exp\left(-\frac{(\mu_1 - (y_1 + \mu_2)/2)^2}{2\sigma^2/2}\right) \quad (\text{this is accorind to the pdf})$$

And for $i = n$, the procedure is totally same, so, we can get Similar to μ_1 , we can compute that :

$$p(\mu_n | \vec{\mu}_{-n}, \vec{Y}) \propto \exp\left(-\frac{(y_n - \mu_n)^2 + (\mu_n - \mu_{n-1})^2}{2\sigma^2}\right)$$

$$\propto \exp\left(-\frac{(\mu_n - (y_n + \mu_{n-1})/2)^2}{2\sigma^2/2}\right),$$

For other part (i not 1 nor n), we use the same theory, and get

$$p(\mu_i | \vec{\mu}_{-i}, \vec{Y}) \propto \exp\left(-\frac{(y_i - \mu_i)^2 + (\mu_{i+1} - \mu_i)^2 + (\mu_i - \mu_{i-1})^2}{2\sigma^2}\right)$$

$$\propto \exp\left(-\frac{(\mu_i - (y_i + \mu_{i-1} + \mu_{i+1})/3)^2}{2\sigma^2/3}\right), \quad i \in (1, n).$$

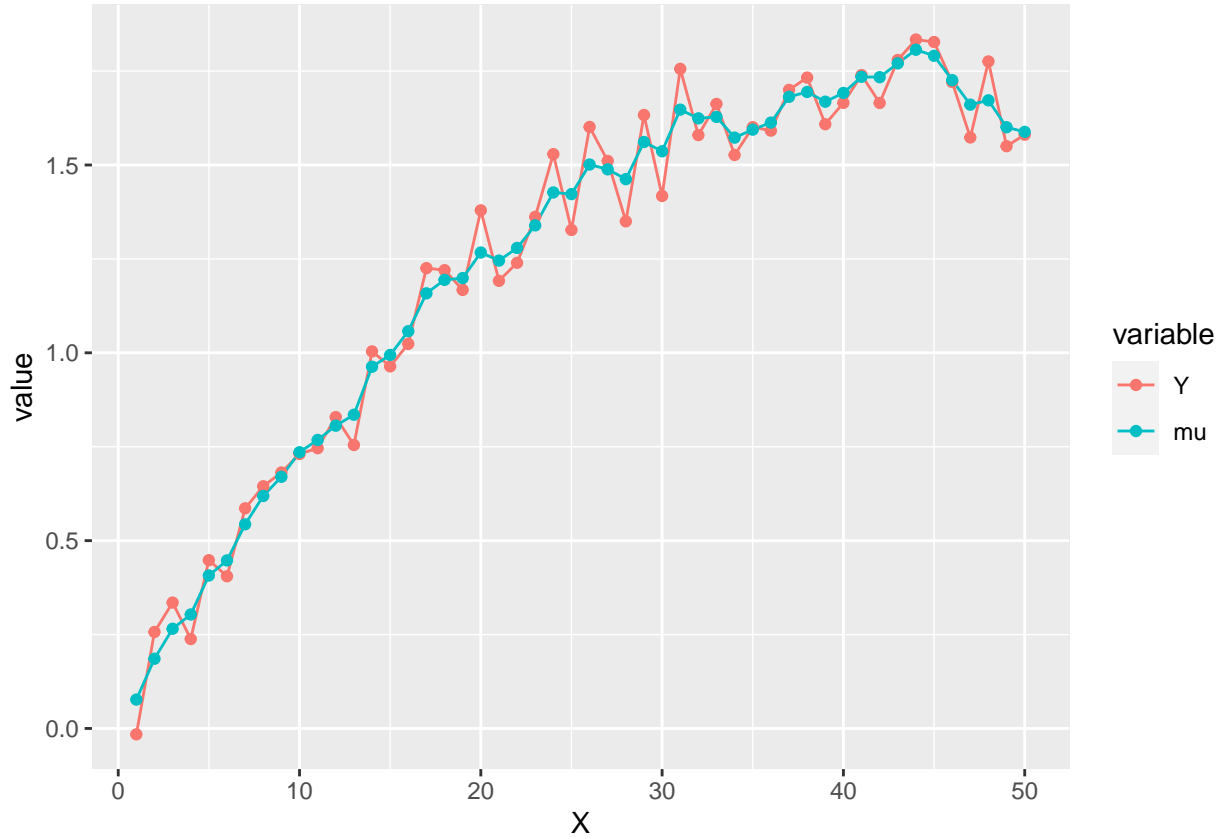
Finally, we find the exp part is similar to normal distribution. So, we conclude that:

$$(\mu_i | \vec{\mu}_{-i}, \vec{Y}) \begin{cases} N(\frac{y_1 + \mu_2}{2}, 0.1) & i = 1 \\ N(\frac{y_i + \mu_{i-1} + \mu_{i+1}}{3}, \frac{0.2}{3}) & \text{Other} \\ N(\frac{y_n + \mu_{n-1}}{2}, 0.1) & i = n \end{cases}$$

Q3: implement Gibbs sampler and Monte Carlo approach.

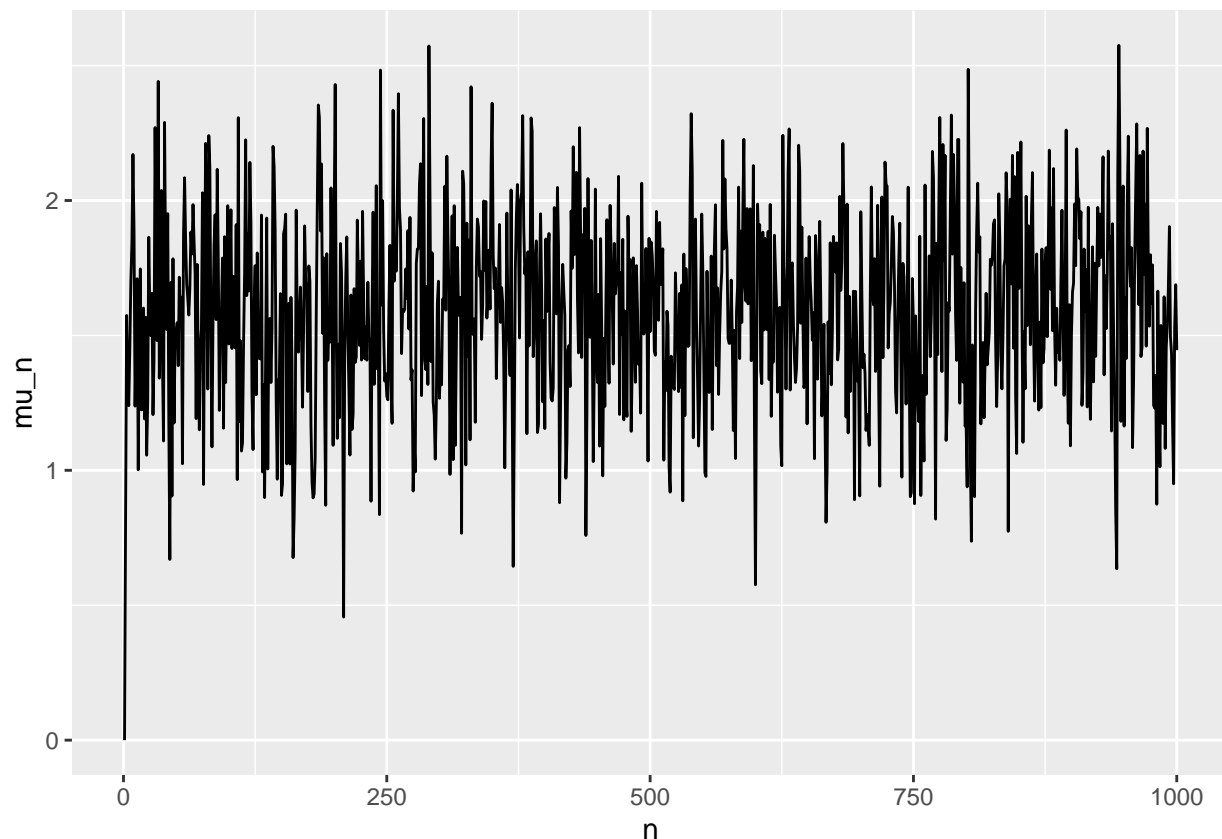
solution:

The curve seems to be more smooth so the noise should be removed. $\vec{\mu}$ can catch underlying dependence between Y and X.



Q4: Make a trace plot for μ_n

solution: burn-in period is very short at the beginning, then the curve converges.



Appendix:

Some code related to output are like ‘## expression’

```
library(coda)
library(ggplot2)
library(reshape2)
#####
#Question 1
#####
#q1
#####
pix <- function(x)
{
  return(x^5*exp(-x))
}
fMH_q1<-function(nstep,X0,props){
  vN<-1:nstep
  vX<-rep(X0,nstep);
  for (i in 2:nstep){
    X<-vX[i-1]
    Y<-rlnorm(1,meanlog=log(X),sdlog =props)
    u<-runif(1)
    a<-min(c(1,(pix(Y)*dlnorm(X,meanlog=log(Y),sdlog=props))/(pix(X)*dlnorm(Y,meanlog=log(X),sdlog=1)))
    if (u <=a){vX[i]<-Y}else{vX[i]<-X}
  }
}
## plot(vN,vX,pch=19,cex=0.3,col="black",xlab="t",ylab="X(t)",main="",ylim=c(min(X0-0.5,0),max(5,X0+0.5)))
```

Inorm是我们用来产生样本的概率函数，pix是目标函数。这里meanlog和sdlog的选取是根据lab pdf里的要求LN(Xt, 1)。X0为初始值，props是额外参数，这里就是那个1。

```

    return(vX)
  }
  t <- 1
  q1_chains <- fMH_q1(10000,1,1)
  rm(t)
#####
# q2
#####
fMH_q2<-function(nstep,X0){
  vN<-1:nstep
  vX<-rep(X0,nstep);
  for (i in 2:nstep){
    X<-vX[i-1]
    Y<-rchisq(1,df=floor(X+1))
    u<-runif(1)
    a<-min(c(1,(pix(Y)*dchisq(X,df=floor(Y+1)))/(pix(X)*dchisq(Y,df=floor(X+1))))
    if (u <=a){vX[i]<-Y}else{vX[i]<-X}
  }
  return(vX)
}
q2_chains <- fMH_q2(10000,1)
## plot(1:10000,q2_chains,pch=19,cex=0.3,col="black",xlab="t",ylab="X(t)",main="",ylim=c(min(1-0.5,0),m
#####
# q3
#####
mc_list <- mcmc.list()
for (i in 1:10)
{
  fmh_vX <- fMH_q2(1000,i)
  mc_list[[i]] <- as.mcmc(fmh_vX)
}
## gelman.diag(mc_list)
#####
# q4
#####
mean_1 <- mean(q1_chains)
mean_2 <- mean(q2_chains)
## print(paste0('for sample from question1: ',mean_1))
## print(paste0('for sample from question1: ',mean_2))
#####
# Q2 Gibbs sampling
#####
#q3
#####
load('chemical.RData')
nstep <- 1000
len <- length(Y)
m0 <- rep(0,len)
gibbs <- function(nstep,Y,m0)
{
  n <- length(m0)
  gbs_matrix_mu <- matrix(0, nrow=nstep, ncol=n)
  gbs_matrix_mu[1,] <- m0

```

这个矩阵每一行是一次采样， $(\mu_1, \mu_2, \dots, \mu_{50})$ ，所以列平均就是期望值。

```
for (t in 2:nstep)
{
  gbs_matrix_mu[t,1] <- rnorm(1,(Y[1]+gbs_matrix_mu[t-1,2])/2,sqrt(0.1))
  for (i in 2:(n-1))
  {
    gbs_matrix_mu[t,i] <- rnorm(1,(Y[i]+gbs_matrix_mu[t,i-1]+gbs_matrix_mu[t-1,i+1])/3,sqrt(0.2/3))
  }
  gbs_matrix_mu[t,n] <- rnorm(1,(Y[n]+gbs_matrix_mu[t,n-1])/2,sqrt(0.1))
}
return(gbs_matrix_mu)
}
sample_mu <- gibbs(nstep = nstep,Y=Y,m0 = m0)
for_plot <- data.frame(X=X,Y=Y,mu=colMeans(sample_mu))
for_plot_melt <- reshape2::melt(for_plot,id='X')
##ggplot(for_plot_melt,aes(x=X,y=value,color=variable))+
##  geom_line()+
##  geom_point()
#####
#q4
#####
plot_mu <- data.frame(n=1:1000,mu_n=sample_mu[,50])
## ggplot(plot_mu,aes(x=n,y=mu_n))+
##  geom_line()
```