99

統算 HW10

統碩一 109225017 林心玹

1

$Y \sim Bin(50,\theta), \theta \sim Beta(0.5,0.5), \pi(\theta|y) \propto \theta^{y-0.5}(1-\theta)^{n-y-0.5} I(0<\theta<1)$

$\theta_{i+1} \sim N(\theta_i, \sigma^2), \sigma = 0.1, 0.5$

N=1000, to get posterior samples: $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(1000)}$

Algorithm:

Step1: Generate $y \sim N(50,0.8)$

Step2: Given initial $\theta_n = 0.5, n = 0$

Step3: Generate $\theta \sim N(\theta_n = 0.5, \sigma^2), 0 < \theta < 1$

Step4: Generate $u \sim U(0,1)$

Step5: $\alpha_{i,i+1} = min(\frac{\pi(\theta_{i+1}|y)}{\pi(\theta_i|y)}, 1) = min((\frac{\theta_{i+1}}{\theta_i})^{y-0.5}(\frac{1-\theta_{i+1}}{1-\theta_i})^{50-y-0.5}, 1)$

If $u < \alpha_{i,i+1}, then \quad \theta_{n+1} = \theta,$

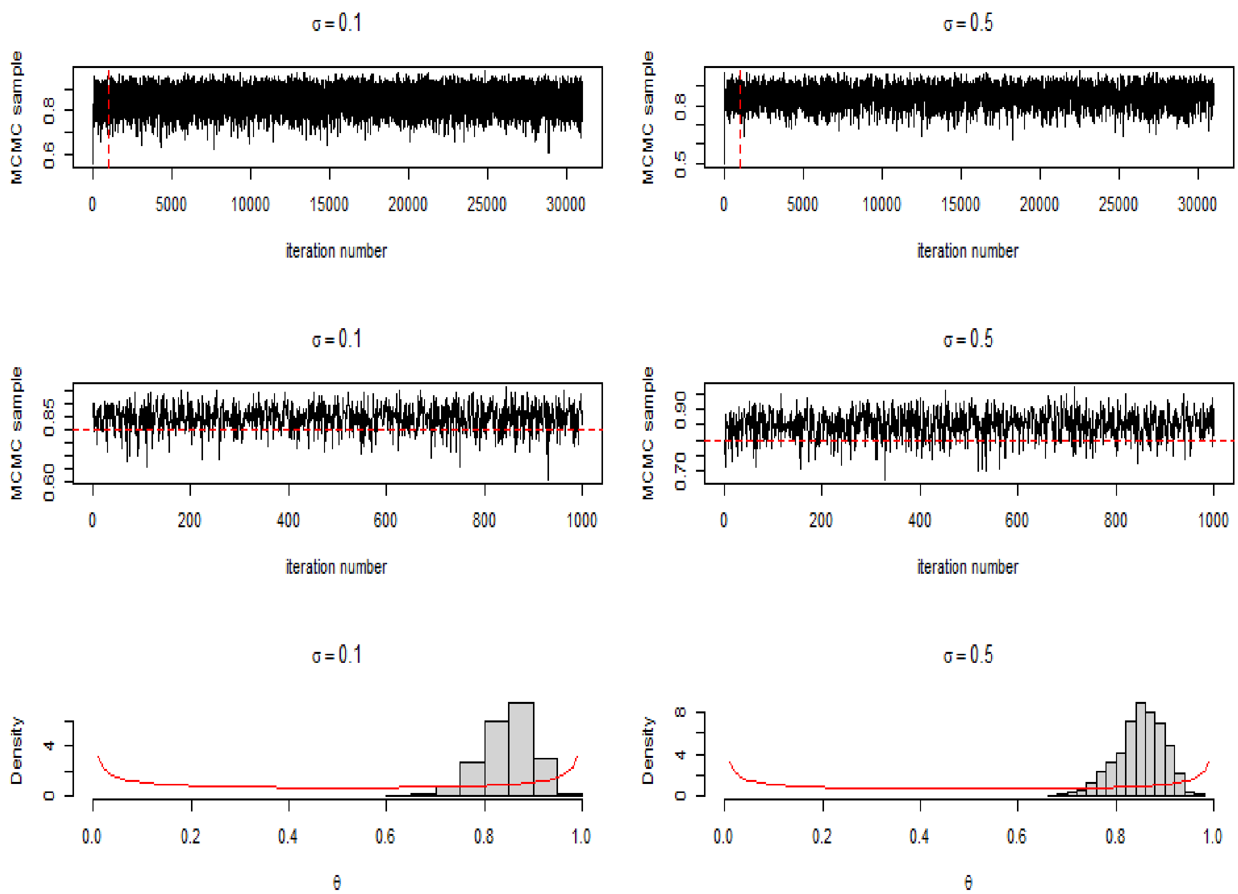
$o.w., \theta_{n+1} = \theta_n$

Step6: n=n+1, $\theta_n = \theta_{n+1}$, return to Step2 $(m + N * k)$ times

$\Rightarrow \theta^{(m+k)}, \theta^{(m+2k)}, \dots, \theta^{(m+1000k)} \sim \pi(\theta|y)$

```
posterior <- function(theta,y){(theta^(y-0.5))*((1-theta)^(50-y-0.5))}
f <- function(m=1000,N=1000,k=30,theta0=0.5,sigma){
  a <-c();theta <- c()
  y0 <- rbinom(1,50,0.8)
  u <- runif(m+N*k)
  for(i in 1:(m+N*k)){
    repeat{
      z <- rnorm(1,theta0,sigma)
      if(z>0 & z<1){break}
    }
    a[i] <- alpha <- min(posterior(z,y0)/posterior(theta0,y0),1)
    if(alpha>u[i]){
      theta[i] <- z
    }else{
```

```
      theta[i] <- theta0
    }
    theta0 <- theta[i]
  }
  return(list(a,u,theta))
}
set.seed(109225017)
result1 <- f(sigma=0.1)
m=1000;N=1000;k=30
result1.theta <- result1[[3]][seq(m+k,m+N*k,k)]
table <- matrix(c(mean(result1.theta),var(result1.theta),sum(result1
[[1]]>result1[[2]])/(m+N*k)),nrow=1)
set.seed(109225017)
result2 <- f(sigma=0.5)
result2.theta <- result2[[3]][seq(m+k,m+N*k,k)]
table <- rbind(table,c(mean(result2.theta),var(result2.theta),sum(resul
t2[[1]]>result2[[2]])/(m+N*k)))
colnames(table) <- c("E(theta|y)","Var(theta|y)","AR")
rownames(table) <- c("sigma=0.1","sigma=0.5")
table <- as.data.frame(table)
par(mfrow=c(3,2))
plot(1:(m+N*k),result1[[3]],type="l",xlab = "iteration number",ylab="MC
MC sample",main=expression
     (sigma==0.1))
abline(v=m,col="red",lty=2)

plot(1:(m+N*k),result2[[3]],type="l",xlab = "iteration number",ylab="MC
MC sample",main=expression
     (sigma==0.5))
abline(v=m,col="red",lty=2)

plot(1:N,result1.theta,type="l",xlab = "iteration number",ylab="MCMC sa
mple",main=expression(sigma==0.1))
abline(h=0.8,col="red",lty=2)

plot(1:N,result2.theta,type="l",xlab = "iteration number",ylab="MCMC sa
mple",main=expression(sigma==0.5))
abline(h=0.8,col="red",lty=2)

hist(result1.theta,freq = F,xlim=c(0,1),xlab=expression(theta),main=exp
ression(sigma==0.1))
curve(dbeta(x,0.5,0.5),add=T,col="red")

hist(result2.theta,freq = F,xlim=c(0,1),xlab=expression(theta),main=exp
ression(sigma==0.5))
curve(dbeta(x,0.5,0.5),add=T,col="red")
```



	E(theta|y)	Var(theta|y)	AR
sigma=0.1	0.8480974	0.0027426	0.5382903
sigma=0.5	0.8502182	0.0022889	0.2091613

由上表可知，σ=0.1 和σ=0.5 的 E(theta|y) 都接近 0.8，但當σ=0.1 時，AR 明顯高於σ=0.5， 因為後驗分配是在有樣本下對先驗分配的修正,但先驗分配假設過於主觀(變異數很小，theta 會收斂至其期望值)造成與後驗分配差異大。

2

Generate a sample of size m=1000 from $f_Y(y)$

m=1000=N;k=30

Algorithm:

Step1: Specify $y_0 = 0.5$

Step2: Generate $x_0 \sim Bin(n, y_0)$

Step3: n=1

Step4: $y_n \sim Beta(x_{n-1} + \alpha, n - x_{n-1} + \beta)$

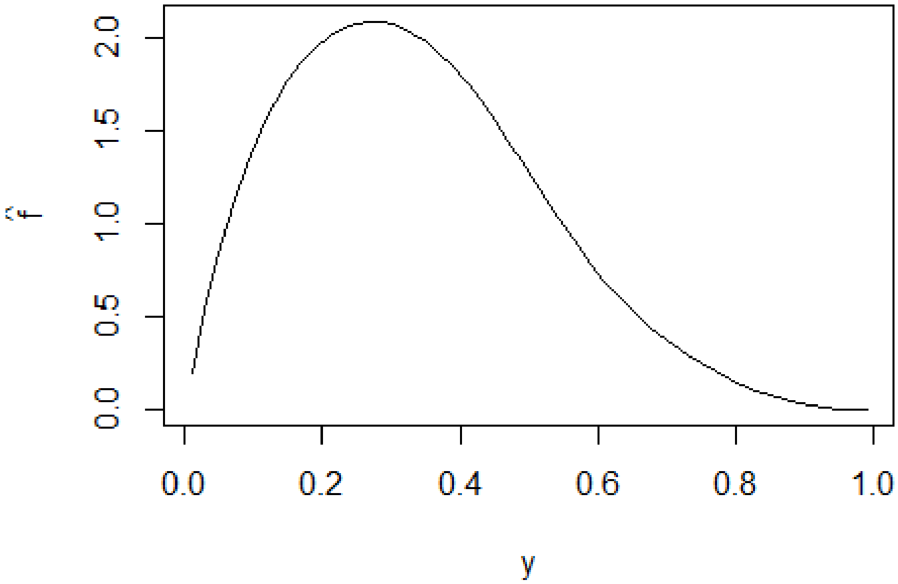
Step5: $x_n \sim Bin(n, y_n)$

Step6: n=n+1, go to Step4 after m+N*k itrations

Step7: Set $x_i^* = x_{m+ik}, \quad y_i^* = y_{m+ik}, \quad i = 1, \dots, N.$

E(Y)≈ $\frac{1}{N}\sum_{i=1}^{N} y_i^*$, Var(Y)≈ $\frac{1}{N-1}\sum_{i=1}^{N}(y_i^* - E(Y))^2$, $\widehat{f_Y}(y) = \frac{1}{N}\sum_{i=1}^{N} f(y|x_i^*), y =$ 0.01,0.03,…,0.99

```r
#2
y0 <- 0.5
N = m = 1000;k <- 30
x <-c(); y <- c()
set.seed(109225017)
for(i in 1:(m+N*k)){
  x[i] <- rbinom(1,20,y0)
  y[i] <- rbeta(1,x[i]+2,20-x[i]+4)
  y0 <- y[i]
}
result <- y[seq(m+k,m+N*k,k)]
xaxis <- seq(0.01,0.99,0.02)
fyhat <- sapply(xaxis, function(y){
  mean(dbeta(y,x[seq(m+k,m+N*k,k)]+2,20-x[seq(m+k,m+N*k,k)]+4))
})
plot(xaxis,fyhat,type = "l",xlab = "y",ylab =expression(hat(f)))
```




```
mean(result);var(result)

## [1] 0.3358946

## [1] 0.03205318
```