Stat651 Mini Project 2

Arthur Lui

8 October 2014

1 Rejection Sampling

1.1 Likelihood Function

I chose a centered and scaled Beta function for my likelihood function. Let $X|a,b \sim Beta(a,b)$, then we transforming the variable X by $Y = 6X + 1$. Then,

$$f_{Y|a,b}(y) = \frac{1}{6B(a,b)}\left(\frac{y-1}{6}\right)^{a-1}\left(\frac{7-y}{6}\right)^{b-1},\ y \in (1,7)$$

where B(a,b) is the beta function.
I chose this as my likelihood function because our data is bounded by 1 and 7, and are continuous.

1.2 Prior Distribution

I chose for my prior distribution,

$$\pi(\theta) = p(\theta_1, 13/4, 10)p(\theta_2, 3.5, 1)$$

where $p(x,a,b) = \frac{1}{\Gamma(a)\Gamma(b)}x^{a-1}e^{-x/b}$. That is, my prior is the joint distribution of two gamma distributions. I chose the values for each prior based on what I believe the expected value of the average scores of the professors should be. Note that I am multiplying the two gamma pdf's because I assume that the two parameters a and b in the likelihood are independent.

1.3 Envelop Function

My envelop function is my prior. It is easy to get draws from this distribution. Also, after scaling, the envelop covers the sampling distribution well.

1.4 Posterior - Rejection Sampling

$$\begin{aligned}
E[\Theta|\boldsymbol{Y}] &= (12.5478, 3.4711)\\
V[\Theta|\boldsymbol{Y}] &= (10.136, 2.3253)\\
SD[\Theta|\boldsymbol{Y}] &= (3.1837, 1.5249)\\
P(\boldsymbol{Y} > 5|\Theta) &= 0.8598
\end{aligned}$$

2 Importance Sampling

2.1 Importance Function

The importance function $I(\Theta)$ I chose was the envelop function I chose for my rejection sampler. That is,

$$I(\Theta) = p(\theta_1, 13/4, 10)p(\theta_2, 3.5, 1)$$

where $p(x,a,b) = \frac{1}{\Gamma(a)\Gamma(b)}x^{a-1}e^{-x/b}$. The importance function I chose is easy to sample from and mirrors the sampling distribution well. It preserves the parameter space because a and b are positive. Also, the gamma distribution has long tails.

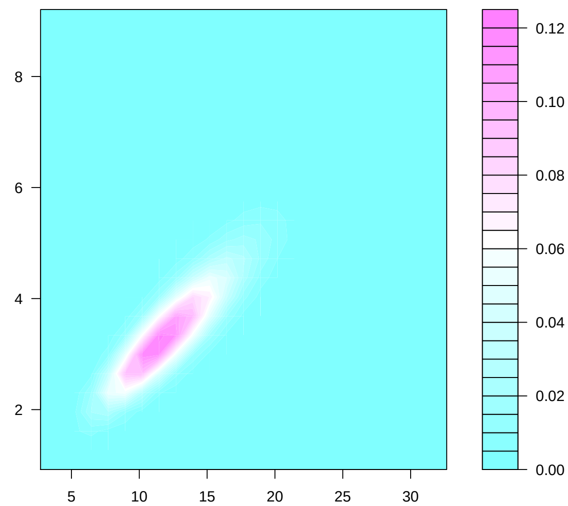

Figure 1: Contour Plot of Posterior Distribution (Rejection Sampling)

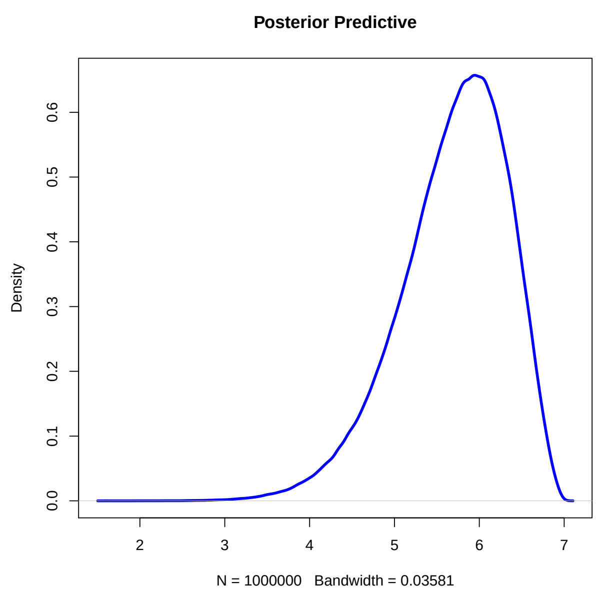

Figure 2: Posterior Predictive Distribution (Rejection Sampling)

2.2 Posterior - Importance Sampling

$$\begin{aligned}
E[\Theta|\boldsymbol{Y}] &= (18.6842, 4.8749) \\
V[\Theta|\boldsymbol{Y}] &= (21.8534, 1.3673) \\
SD[\Theta|\boldsymbol{Y}] &= (4.6748, 1.1693) \\
\hat{c} &= 3.05370998937396e-08
\end{aligned}$$

3 Compare & Contrasts

A rejection sampler is more cumbersome to write when compared to an importance sampler. Deciding on an envelop function can be tedious and difficult especially when the parameter is multi-dimensional. An importance sampler is to write as it only requires an importance function which can be chosen relatively easily, as fewer constraints are imposed on it.

Rejection sampling is slower than importance sampling, especially in the case where the envelop does not tightly envelop the sampling distribution. However, rejection sampling returns the posterior of the parameters, whereas importance sampling only returns estimates of the parameters. Also, in importance sampling, accuracy of the estimates increase with the number of iterations. They may not be as accurate for the same number of draws as rejection sampling.

4 Appendix - Code

```
# Arthur Lui
# MP 2: Code
# 1,.., 7
n <- 1e6

library(doMC); registerDoMC(strtoi(system("nproc",intern=T))/2)

boot <- function(dat,fn,B=1e4,parallel=T) {
  n <- length(as.vector(dat))
  samp <- function() sample(dat,n,replace=T)
  do.it <- function(x) fn(samp())

  if (parallel) {
    btstrp <- foreach(b=1:B,.combine=rbind) %dopar% do.it(i)
  } else {
    btstrp <- apply(matrix(1:B),1, do.it)
  }

  est <- mean(btstrp)
  se <- sd(btstrp)

  out <- matrix(c(est,qnorm(c(.025,.975),est,se)),nrow=1)
  colnames(out) <- c("Estimate","CI.Lower","CI.Upper")
  out
}


source("rejection-sampler.R")
```

```
pb <- function(i,n) {
  cat(paste0("\rProgress: ",round(i*1000/n)/10,"%"))
  if (i==n) cat("\n")
}

fac <- as.vector(as.matrix(read.table("faculty.dat",header=F)))

# Let X|a,b ~ Beta(a,b)
# Y = 6X+1
# f(y) = G(a+b)/(G(a)G(b)) ((y-1)/6)^(a-1) ((7-y)/6)^(b-1) / 6, y el (1,7)
# a ~ G(2,2)
# b ~ G(2,2)
                                                        #
# I believe that mean is in the middle. 4/8 => 4 (1,2,3,4,5,6,7)

f <- function(y,a,b,log=F) {
  out <- NULL
  if (!log) {
    out <- 1 / beta(a,b) * ((y-1)/6)^(a-1) * ((7-y)/6)^(b-1) / 6
  } else {
    out <- -lbeta(a,b) + (a-1)*(log(y-1)-log(6)) + (b-1)*(log(7-y)-log(6)) - log(6)
  }
  out
}

ran.f <- function(n,a,b) {
  6*rbeta(n,a,b)+1
}


#f <- function(y,a,b,log=F) dbeta(y,a,b,log=log)
#ran.f <- function(n,a,b) rbeta(n,a,b)
#fac <- (fac - 1)/6

a1 <- 26/12#26
b1 <- 10#.5
a2 <- 3.5
b2 <- 1

g <- function(x,a,b,log=F) {
  out <- NULL
  if (!log) {
    out <- prod(f(x,a,b)) * dgamma(a,a1,scale=b1) * dgamma(b,a2,scale=b2)
  } else {
    out <- sum(f(x,a,b,log=T)) + dgamma(a,2,scale=2,log=T) + dgamma(b,a2,scale=b2,log=T)
  }
  out
}

log.g <- function(x) g(fac,x[1],x[2],log=T)
new.g <- function(x) exp(log.g(x))
```

```
e <- function(x,log=F) { # Envelop function
  out <- NULL

  if (!log) {
    out <- dgamma(x[1],a1,scale=b1) * dgamma(x[2],a2,scale=b2)
  } else {
    out <- dgamma(x[1],a1,scale=b1,log=T) + dgamma(x[2],a2,scale=b2,log=T)
  }
  out
}

log.e <- function(x) e(x,log=T)
e.sampler <- function() c(rgamma(1,a1,scale=b1),rgamma(1,a2,scale=b2))

x <- seq(.00001,25,length=100)
y <- seq(.00001, 7,length=100)
w <- expand.grid(x,y)
z1 <- apply(w,1,e)
z2 <- apply(w,1,new.g)

library(rgl)
a <- 1/max(z2/z1)
persp3d(x=x,y=y,z=z2,col='blue')
persp3d(x=x,y=y,z=z1/a,col='red',add=T)

X <- rejection.sampler(log.g,log.e,e.sampler,a,n)

library(MASS)
K <- kde2d(X[,1],X[,2])
persp3d(K,col="yellow")
persp3d(x=x,y=y,z=z2,col='blue',add=T)
title3d("Posterior")

pdf("rejPostContour.pdf")
  filled.contour(K)
dev.off()

#3 E[T|Y]:
#(rej.post.mean <- t(apply(X,2,function(x) boot(x,mean))))
(rej.post.mean <- apply(X,2,mean))
#4 V[T|Y]:
(rej.post.var <- var(X))
#5 SD[T|Y]:
(rej.post.sd <- sqrt(rej.post.var))
#6 Pred:
M <- apply(X,1,function(x) ran.f(1,x[1],x[2]))
plot(density(fac),col="red",lwd=3)
lines(density(M),col="blue",lwd=3)
pdf("postPred.pdf")
  plot(density(M),col="blue",lwd=3,main="Posterior Predictive")
```

```
dev.off()

#7
(prob.greater.than.5 <- mean(M>5))

#Importance: ###########################3
#1:
# I ~ Gamma(a1,b1) * Gamma(a2,b2)

importance.sampling <- function(h=function(x) x,B=n,get.c=F){
  I <- function(p) e(p,log=F)
  P <- t(apply(matrix(1:B),1,function(x) e.sampler()))

  num.fun <- function(p) h(p) * g(fac,p[1],p[2]) / I(p)
  den.fun <- function(p) g(fac,p[1],p[2]) / I(p)
  comp.fun <- function(p) c(num.fun(p), den.fun(p))

  Y <- t(apply(P,1,comp.fun))
  mean.a <- sum(Y[,1]) / sum(Y[,3])
  mean.b <- sum(Y[,2]) / sum(Y[,3])

  a.dat <- Y[,1]/mean(Y[,3])
  b.dat <- Y[,2]/mean(Y[,3])
  CI.a <- qnorm(c(.025,.975),mean(a.dat),sd(a.dat)/sqrt(B))
  CI.b <- qnorm(c(.025,.975),mean(b.dat),sd(b.dat)/sqrt(B))

  result <- matrix(c(mean.a,CI.a[1],CI.a[2],
                     mean.b,CI.b[1],CI.b[2]),nrow=2,byrow=T)
  colnames(result) <- c("Estimate","CI Lower","CI Upper")
  rownames(result) <- c("a","b")

  if (get.c) {
    result <- list("result"=result,"normalizing.constant"=mean(Y[,3]))
  }

  result
}

#2 E(P|Y):


temp <- importance.sampling(B=n,get.c=T)
imp.post.mean <- temp[[1]]
norm.const <- temp[[2]]
#3 V(P|Y):
imp.post.var <- importance.sampling(function(x) (x-imp.post.mean[,1])^2, B=n)
#4 SD(P|Y):
imp.post.sd <- sqrt(imp.post.var)
#5 C:
norm.const
```