STAT 651 Mini Project 2

Stephen Merrill

February 11, 2017

1 Likelihood Function

I used a four parameter Beta function in order to align the support of the likelihood with the support of the data. I believe this distribution can appropriately capture probable data values, and can do so over the correct range of possible values (1-7). The pdf of this distribution is given as follows:

$$f(y; \alpha, \beta, a, c) = \frac{f(x; \alpha, \beta)}{c - a} = \frac{\left(\frac{y-a}{c-a}\right)^{\alpha-1}\left(\frac{c-y}{c-a}\right)^{\beta-1}}{(c-a)B(\alpha, \beta)} = \frac{(y-a)^{\alpha-1}(c-y)^{\beta-1}}{(c-a)^{\alpha+\beta-1}B(\alpha, \beta)}$$

Where c is fixed at 7 and a is fixed at 1. This is the pdf, and the likelihood would of course be the product of the pdf over the data values. A possible visualization of the likelihood can be seen in Figure 1.
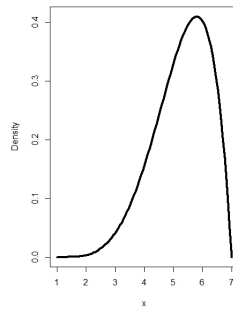

Figure 1: This four-parameter Beta Distribution, for example, represents my belief of how the rating data could be distributed.

2 Prior Distributions

Since the support of α and β is greater than zero, I used a gamma distribution for both parameters. The shape and scale values were selected with consideration to the equations for the mean and variance of the beta likelihood. I believed the mean of teacher ratings at BYU would be above average, with a somewhat substantial amount of variance, and my choices for prior parameters values reflected that belief. These distributions are displayed in Figure 2.

$$\alpha \sim Gamma(shape = 0.5, scale = 16)$$
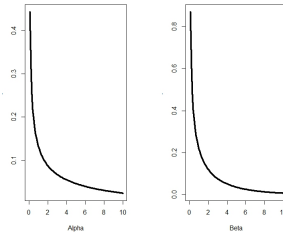$$\beta \sim Gamma(shape = 0.5, scale = 4)$$

Figure 2: Prior Distributions

3 Rejection Sampling

3.1 Envelope Function

I used a multivariate normal envelope function because of its ability to cover all values of the unnormalized posterior $g(\theta)$. This can be seen in figure 3. Specifically, this distribution is:

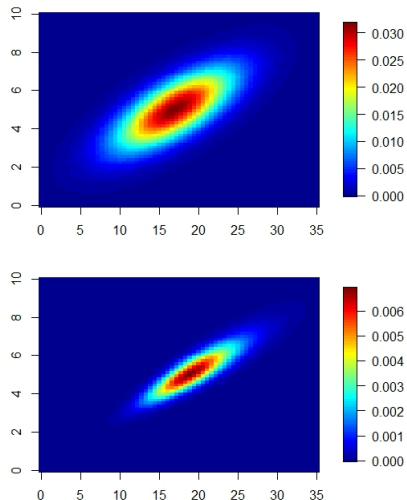$$MVN\left(\begin{bmatrix} 17 \\ 5 \end{bmatrix}, \begin{bmatrix} 25 & 5 \\ 5 & 2 \end{bmatrix} \right)$$

Figure 3: The envelope above covers all values of $g(\theta)$, which as been multiplied by a constant to improve the performance of the rejection sampler.

3.2 Posterior Distribution

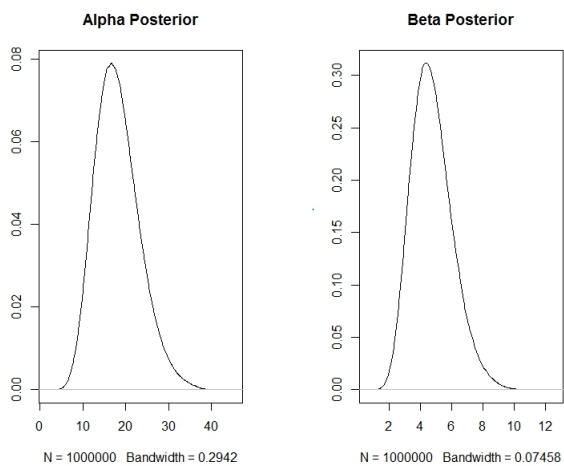Figure 4 shows the posterior distributions for α and β using 1,000,000 draws from the rejection sampler.

Figure 4: Posterior Distributions

3.3 Predictive Distribution

Figure 5 shows the predictive distribution for the next score.

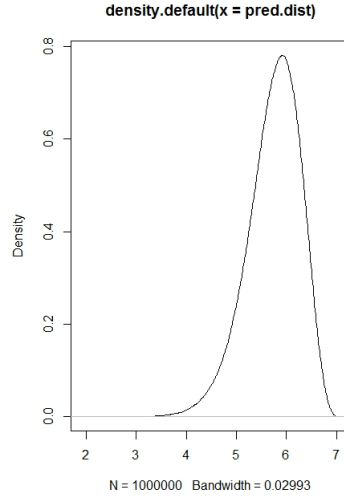**density.default(x = pred.dist)**

N = 1000000   Bandwidth = 0.02993

Figure 5: Predictive Distribution

3.4 Summary Statistics

- Mean, $E(\theta|Y) = \begin{bmatrix} 18.0611 \\ 4.7584 \end{bmatrix}$

- Variance, $V(\theta|Y) = \begin{bmatrix} 26.8848 & 6.3629 \\ 6.3629 & 1.7359 \end{bmatrix}$

- Standard Deviation, $\sqrt{V(\theta|Y)} = \begin{bmatrix} 5.1850 \\ 1.3175 \end{bmatrix}$

- Probability next score is greater than 5, $P(\tilde{Y} > 5) = .9082$

4 Importance Sampling

4.1 Importance Function

I used the same multivariate normal function that I used as my envelope as my importance function. This function works well as $I(\theta)$ because it mimics the values of the likelihood, as seen in Figure 3, and it covers the tails well.

4.2 Summary Statistics

- Mean, $E(\theta|Y) = \begin{bmatrix} 18.0689 \\ 4.7603 \end{bmatrix}$

- Variance, $V(\theta|Y) = \begin{bmatrix} 27.0751 \\ 1.7954 \end{bmatrix}$

- Standard Deviation, $\sqrt{V(\theta|Y)} = \begin{bmatrix} 5.2033 \\ 1.3399 \end{bmatrix}$

4.3 Normalizing Constant

The normalizing constant was calculated as 3.902e-09. This can be interpreted as the value that $g(\theta)$ is to be divided by to integrate to one and be a density.

5 Comparison and Contrast

Importance sampling is much less computationally intensive than rejection sampling in terms of computing time. In terms of the estimates produced, the importance sampling matches up closely. The rejection sampler was more difficult to implement, mostly because of the struggle to appropriately envelope the unnormalized posterior.

6 Appendix: Code

```
score<-read.table("http://madison.byu.edu/bayes/faculty.dat")
library(ExtDist)
par(mfrow=c(1,1))
curve(dBeta_ab(x,5,2,1,7),xlim=c(1,7),lwd=4,ylab="Density")
par(mfrow=c(1,2))
curve(dgamma(x,.5,scale=16),xlim=c(0,10),lwd=4,xlab="Alpha",ylab="Density")
curve(dgamma(x,.5,scale=4),xlim=c(0,10),lwd=4,xlab="Beta",ylab="Density")

g <-function(a,b) {
   prod(dBeta_ab(score[,1],a,b,1,7))*dgamma(a,.5,scale=16)*dgamma(b,.5,scale=4)
}
library(mvtnorm)
e<-function(a,b) {
   dmvnorm(x=c(a,b),mean=c(17,5),sigma=matrix(c(25,5,5,2),nrow=2))
}

a<-seq(.1,35,by=.1)
b<-seq(.1,9,by=.1)
z<-matrix(NA,nrow=length(a)*length(b),ncol=4)

#verify envelope coverage

for (i in 1:length(a)) {
   for(j in 1:length(b)) {
      z[j+length(b)*i-length(b),1]<-a[i]
      z[j+length(b)*i-length(b),2]<-b[j]
      z[j+length(b)*i-length(b),3]<-g(a[i],b[j])*1e+07
      z[j+length(b)*i-length(b),4]<-e(a[i],b[j])
   }
}
sum(z[,3]>z[,4])

#rejection sampler draws

M<-1000000
theta<-matrix(NA,ncol=2,nrow=M)
m=0
niter<-0
```

```r
while(m<M) {
  xy<- c(-1,-1)
  while(xy[1] < 0 | xy[2] < 0) {
    xy<-rmvnorm(1,mean=c(17,5),sigma=matrix(c(25,5,5,2),nrow=2))
  }
  u<-runif(1)
  r<-g(xy[1],xy[2])*1e+07/e(xy[1],xy[2])
  if(u<r) {
    m<-m+1
    theta[m,1]<-xy[1]
    theta[m,2]<-xy[2]
  }
  niter<-niter+1
}

plot(density(theta[,1]),main="Alpha Posterior")
plot(density(theta[,2]),main="Beta Posterior")

#summary statistics

a.star <- mean(theta[,1])
b.star <- mean(theta[,2])
var.post <- var(theta)
sd.post <- sqrt(var.post)

#predictive distribution

pred.dist <- numeric(M)
for(i in 1:M) {
  index <- sample(1:M,1)
  a.samp <- theta[index,1]
  b.samp <- theta[index,2]
  pred.dist[i] <- rBeta_ab(1,a.samp,b.samp,1,7)
}

plot(density(pred.dist))

#prob > 5
length(which(pred.dist>=5))/M

library(LatticeKrig)
par(mfrow=c(2,1))
quilt.plot(x=z[,1],y=z[,2],z=z[,4])
quilt.plot(x=z[,1],y=z[,2],z=z[,3])

#Importance sampling

I<-function(a,b) {
  dmvnorm(x=c(a,b),mean=c(17,5),sigma=matrix(c(25,5,5,2),nrow=2))
}

h<-function(theta) {
  theta
}
```

```
#sampler

M<-1000000
draws<-matrix(NA,nrow=M,ncol=2)
i<-0
while(i<M)  {
   draw<-rmvnorm(1,mean=c(17,5),sigma=matrix(c(25,5,5,2),nrow=2))
   if(draw[1,1]>0 & draw[1,2]>0) {
     i<-i+1
     draws[i,]<-draw
   }
}

num.alpha<-numeric(M)
for(i in 1:M){
   num.alpha[i]<-h(draws[i,1])*g(draws[i,1],draws[i,2])/I(draws[i,1],draws[i,2])
}

den<-numeric(M)
for(i in 1:M) {
   den[i]<-mean(g(draws[i,1],draws[i,2])/I(draws[i,1],draws[i,2]))
}

alpha.m<-mean(num.alpha)/mean(den)

num.beta<-numeric(M)
for(i in 1:M){
   num.beta[i]<-h(draws[i,2])*g(draws[i,1],draws[i,2])/I(draws[i,1],draws[i,2])
}

beta.m<-mean(num.beta)/mean(den)

#mean(den) == c

#2nd moment

h<-function(theta) {
   theta^2
}
num.alpha2<-numeric(M)
for(i in 1:M){
   num.alpha2[i]<-h(draws[i,1])*g(draws[i,1],draws[i,2])/I(draws[i,1],draws[i,2])
}
alpha.2<-mean(num.alpha2)/mean(den)
num.beta2<-numeric(M)
for(i in 1:M){
   num.beta2[i]<-h(draws[i,2])*g(draws[i,1],draws[i,2])/I(draws[i,1],draws[i,2])
}
beta.2<-mean(num.beta2)/mean(den)
var.alpha<- alpha.2-alpha.m^2
var.beta<- beta2-beta.m^2
```