MA 578: Bayesian HW4

Benjamin Draves

10/28/2019

Exercise 1

Exercise 2.11 (a)

Suppose that we have $y_1, y_2, \ldots, y_5 | \theta \sim \text{Cauchy}(\theta, \gamma = 1)$ and $\theta \sim \text{Unif}(-4, 4)$. We first look to calculate the unnormalized posterior density function

$$p(\theta|y) \propto p(\theta)p(y|\theta) \propto \prod_{i=1}^{5} \frac{1}{1 + (y_i - \theta)^2}$$

We implement this procedure below.

```
#read in data
y <- c(-2,-1,0,1.5,2.5)

#set up grid
m <- 1000
theta_grid <- seq(-4, 4, length.out = m)

#set up unormalized density function
f_unnormalized <- function(theta) prod(sapply(y, function(x) 1/(1 + (x - theta)^2)))

#calculate unnormalized density
unnorm_dens <- sapply(theta_grid, f_unnormalized)
```

To plot the exact posterior distribution, we first need to normalize this distribution. Notice that as $p(\theta) = 1/8$ we have

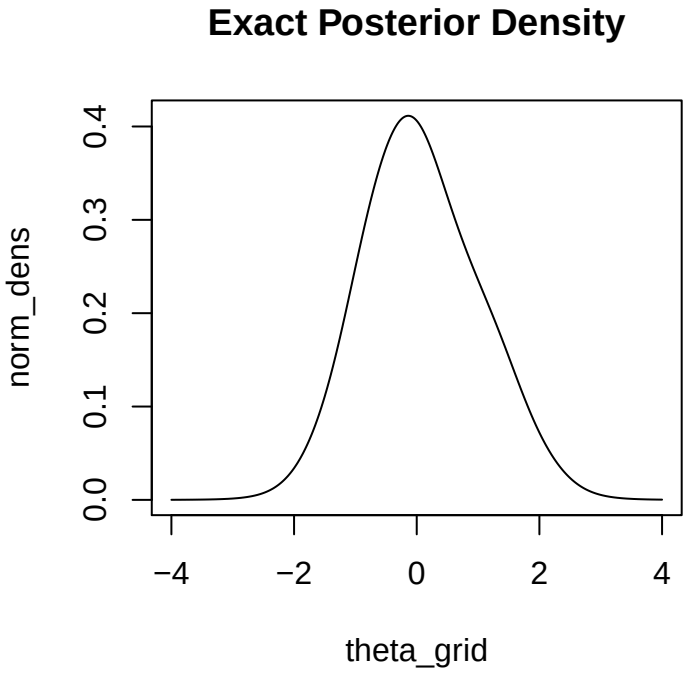
$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)} = \frac{p(\theta)p(y|\theta)}{\int p(\theta)p(y|\theta)d\theta} = \frac{p(y|\theta)}{\int p(y|\theta)d\theta}$$

Above we calculated $p(y|\theta)$ for $\theta^{(s)} \in \Theta$ where Θ was a grid of values between $[-4, 4]$. Moreover, we approximate this integral by

$$\int p(y|\theta)d\theta \approx \Delta \sum_{\theta \in \Theta} p(y|\theta) = \frac{8}{m} \sum_{\theta \in \Theta} p(y|\theta)$$

where $\Delta = 8/m$. Using this we can calculate the normalized distribution as follows.

```
#calculate normalized density
norm_dens <- unnorm_dens/sum(8/m * unnorm_dens)
#plot density
plot(theta_grid, norm_dens, type = "l", main = "Exact Posterior Density")
```



Exercise 4.1

Suppose that $y_1, \ldots, y_5|\theta \sim \text{Cauchy}(\theta, \gamma = 1)$ so that the likelihood is given by

$$p(y_i|\theta) \propto \frac{1}{1 + (y_i - \theta)^2}$$

Assume that $\theta \sim \text{Unif}(0, 1)$.

(a)

First notice as $p(\theta) \propto 1$ that the posterior density is given by

$$p(\theta|y) \propto p(y|\theta)p(\theta) \propto \prod_{i=1}^{5} p(y_i|\theta)$$

Therefore the log posterior is equivalent to the log likelihood. We can write the full log-likelihood as

$$\ell(\theta) = \log\left[p(y_1, \ldots, y_5|\theta)\right] = -\sum_{i=1}^{5} \log(1 + (y_i - \theta)^2)$$

Using this expression we can find its first and second derivative as of the the log likelihood as

$$
\begin{aligned}
\ell'(\theta) &= \sum_{i=1}^{5} \frac{2(y_i - \theta)}{1 + (y_i - \theta)^2} \\
\ell''(\theta) &= \sum_{i=1}^{5} \frac{(-2)(1 + (y_i - \theta)^2 - 2(y_i - \theta)(-2(y_i - \theta))}{[1 + (y_i - \theta)^2]^2} \\
&= \sum_{i=1}^{5} \frac{-2(1 + (y_i - \theta)^2) + 4(y_i - \theta)^2}{[1 + (y_i - \theta)^2]^2} \\
&= 2\sum_{i=1}^{5} \frac{(y_i - \theta)^2 - 1}{[1 + (y_i - \theta)^2]^2}
\end{aligned}
$$

(b)

It appears that finding the solution to $\ell'(\theta) = 0$ will be intractable in this setting. For this reason, to find the posterior mode we use a Newton-Raphson update to iteratively solve the root finding problem $\ell'(\theta) = 0$. After random initialization, the Networn-Raphson updates take the form

$$
\theta^{(t+1)} = \theta^{(t)} - \frac{\ell'(\theta)}{\ell''(\theta)}
$$

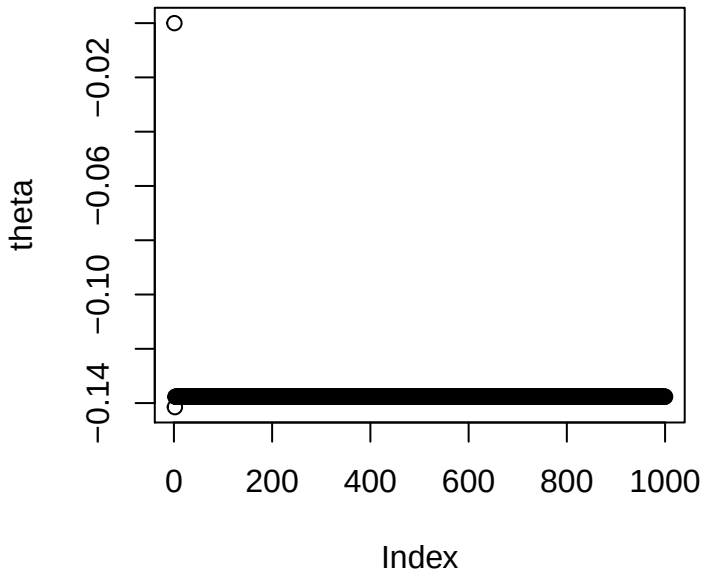
Repeating this update for large values of t, we expect to see $\lim_{t\to\infty} \theta^{(t)} = \hat{\theta}_{mode}$. We implement this update below.

```r
#read in data
y <- c(-2,-1,0,1.5,2.5)

#define l' and l''
l_prime <- function(theta){
  return(sum(sapply(y, function(x) 2*(x - theta)/(1 + (x - theta)^2))))
}
l_double_prime <- function(theta){
  return(2 * sum(sapply(y, function(x) ((x - theta)^2 - 1)/(1 + (x - theta)^2)^2)))
}

#initialize updates
no.iters <- 1000
theta <- numeric(no.iters)

#iteraively solve
for(t in 2:no.iters){
  theta[t] <- theta[t - 1] - l_prime(theta[t-1])/l_double_prime(theta[t-1])
}
plot(theta)
```



It appears that the Newton Raphson update converges rapidly to $\hat{\theta}_{mode}$ =-0.1376493.

(c)

Recall from Laplace's Approximation that for sufficiently large n

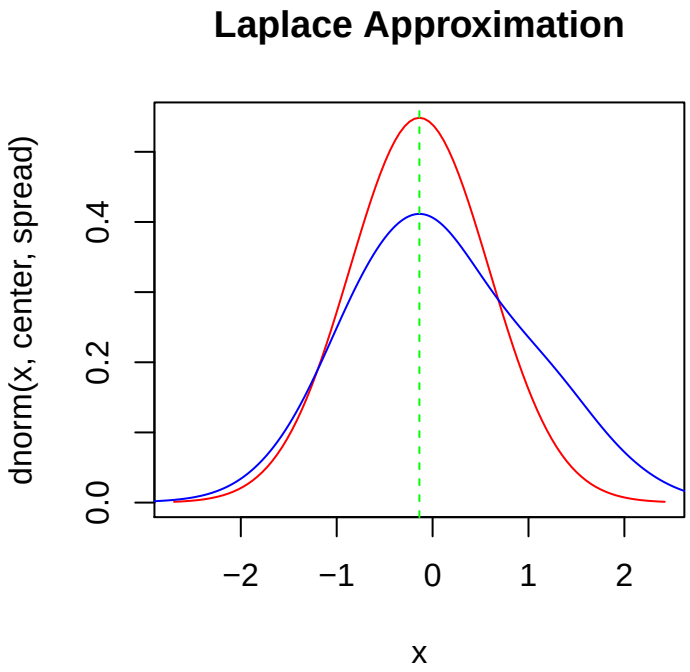
$$\theta|y \approx N(\hat{\theta}_{mode}, I_{\text{obs}}^{-1}(\hat{\theta}_{mode}))$$

where $I_{\text{obs}}^{-1}(\hat{\theta}_{mode}) = -\ell''(\hat{\theta}_{mode})$. We plot this distribution below (red) and compare it to the exact distribution (blue) calculated in 2.11.

```
#set theta_star
theta_star <- theta[no.iters]

#get mean and variance
center <- theta_star
spread <- -1/l_double_prime(theta_star)

#plot curve of normal distribution
curve(dnorm(x, center, spread),
      from = theta_star - 3 * sqrt(spread), to = theta_star + 3 * sqrt(spread),
      col = "red", main = "Laplace Approximation")
points(theta_grid, norm_dens, type = "l", col = "blue")
abline(v = center, col = "green", lty = 2)
```



From here we can see that the normal approximation centers around the same mode as that of the exact distribution. However, the exact distribution appears to have more of a right tail than that of the approximation. While the left tail is roughly equivalent, the right tail is the only major difference between these two distributions. As n increases past $n = 5$ we expect this approximation to become more and more accurate.

Exercise 2

BDA 5.13

(a)

Suppose that $\{y_j\}_{j=1}^{10}$ are the observed number of bikes on residential roads with a bike route. Let $\{n_j\}_{j=1}^{10}$ be the number of vehicles observed on the residential roads with bike routes. Then a reasonable model to consider would be of the form

$$y_j|\theta_j \sim \text{Binom}(n_j, \theta_j)$$

To invoke a hierarchical structure to the model we further assume that

$$\theta_j|\alpha, \beta \sim \text{Beta}(\alpha, \beta)$$
$$p(\alpha, \beta) \propto (\alpha + \beta)^{-5/2}$$

Together the joint posterior density can be given by

$$p(\theta, \alpha, \beta|y) \propto p(y|\theta)p(\theta|\alpha, \beta)p(\alpha, \beta) \propto (\alpha + \beta)^{-5/2} \prod_{j=1}^{10} \theta_j^{y_j}(1 - \theta_j)^{n_j - y_j} \prod_{j=1}^{10} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha-1}(1 - \theta_j)^{\beta-1}$$

(b)

Following a similar analysis as section 5.3 we first note that

$$p(\theta|\alpha,\beta,y) = \prod_{j=1}^{10} \frac{\Gamma(\alpha+\beta+n_j)}{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}\theta^{\alpha+y_j-1}(1-\theta)^{\beta+n_j-y_j-1}$$

Next notice that

$$p(\theta,\alpha,\beta|y) = p(\theta|\alpha,\beta,y)p(\alpha,\beta|y)$$

which upon rearranging gives

$$p(\alpha,\beta|y) = \frac{p(\theta,\alpha,\beta|y)}{p(\theta|\alpha,\beta,y)}$$

Plugging in these values, we have the following

$$p(\alpha,\beta|y) \propto (\alpha+\beta)^{-5/2} \prod_{j=1}^{10} \frac{\Gamma(\alpha+\beta)\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(\alpha+\beta+n_j)}$$

Using this posterior marginal, we can first draw samples of $\alpha,\beta|y$. To do so, we layout a grid around the ML estimates of (α,β) and calculate $\log p(\alpha,\beta|y)$. After this we can sample the parameters and hyperparmeters from the posterior distribution by

1. Simulation B draws from $(\alpha,\beta)^{(b)} \sim p(\alpha,\beta|y)$ for $b = 1,2,\ldots,B$
2. For $b = 1,2,\ldots,B$ and for $j = 1,2,\ldots,10$ sample $\theta_j^{(b)}|(\alpha,\beta)^{(b)},y \sim \text{Beta}(\alpha^{(b)}+y_j, \beta^{(b)}+n_j-y_j)$.

```r
#read in data
ys <- c(16,9,10,13,19,20,18,17,35,55)
ns <- c(58,90,48,57,103,57,86,112,273,64) + ys
dat <- data.frame(y = ys, n = ns)

#set up log prior
lprior <- function(a, b) -2.5 *log(a + b)

#set log posterior
lhood_ab <- function (a, b)
  with(dat,
    sum(lgamma(a + y) - lgamma(a) + lgamma(b + n - y) - lgamma(b) -
        (lgamma(a + b + n) - lgamma(a + b))))


#set up grid around MLE points as we did in class
r <- with(dat, sum(y)/sum(n))
m <- 100
k <- seq(10,30, length = m)
ks <- k[which.max(sapply(k, function (x) lhood_ab(r * x, (1 - r) * x)))]
```

```r
am <- r * ks; bm <- (1 - r) * ks
a <- seq(0, 2 * am, length = m + 1)[-1]
b <- seq(0, 2 * bm, length = m + 1)[-1]

#get log posterior draws
lab <- matrix(nrow = m, ncol = m)
for (ia in 1:m)
  for (ib in 1:m)
    lab[ia, ib] <- lprior(a[ia], b[ib]) + lhood_ab(a[ia], b[ib])

#normalize to get probability grid
pab <- exp(lab - log(sum(exp(lab))))

# sampling alpha, beta functions
pa <- rowSums(pab)
ind_ab <- matrix(1:(m ^ 2), nrow = m)
sample_ab <- function (ns) {
  is <- sapply(sample.int(m ^ 2, ns, replace = TRUE, prob = pab),
               function (ind) which(ind_ab == ind, arr = TRUE))
  cbind(a[is[1,]], b[is[2,]])
}

#sample alpha and betas
B <- 1000
ab_s <- sample_ab(B)

#sample thetas
theta_s <- matrix(nrow = B, ncol = nrow(dat))
for (j in 1:nrow(dat)) {
  theta_s[,j] <- with(dat, rbeta(B, ab_s[,1] + y[j], ab_s[,2] + n[j] - y[j]))
}
```

(c)

Next we look to compare the samples θ values to that of the observed counts. In the plot below, we plot a violin plot to summarize the posterior draws of each θ_j. In addition, we add a red dot to indicate the observed proportion.

```r
library(reshape2); library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

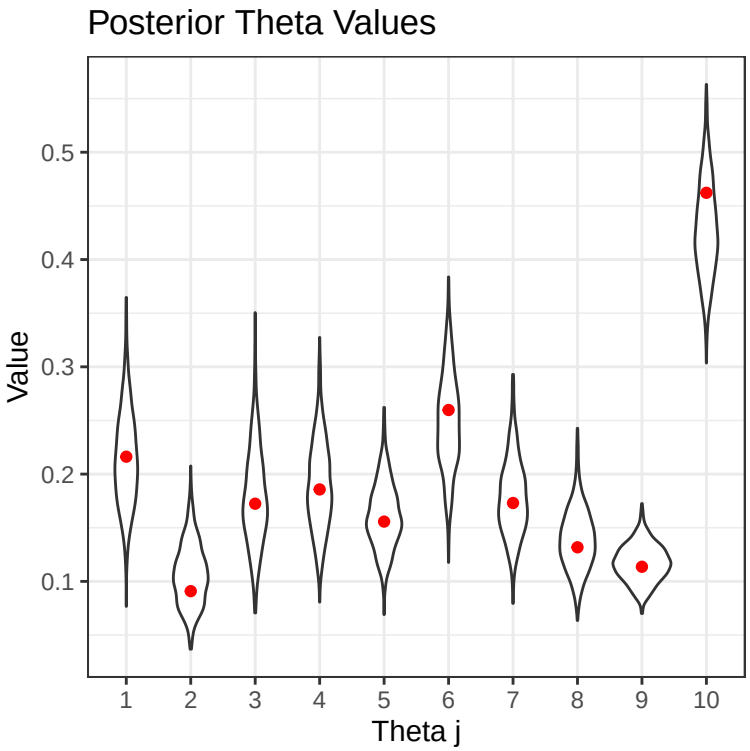
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
colnames(theta_s) <- 1:10
plotdf <- as.data.frame(theta_s) %>%
  melt()
```

```
## No id variables; using all as measure variables
```

```
library(ggplot2)
ggplot(plotdf, aes(variable, value))+
  geom_violin()+
  geom_point(aes(x, obs),data.frame(x = 1:10,obs = dat$y / dat$n), col = "red")+
  theme_bw()+
  labs(x = "Theta j", y = "Value", title = "Posterior Theta Values")
```



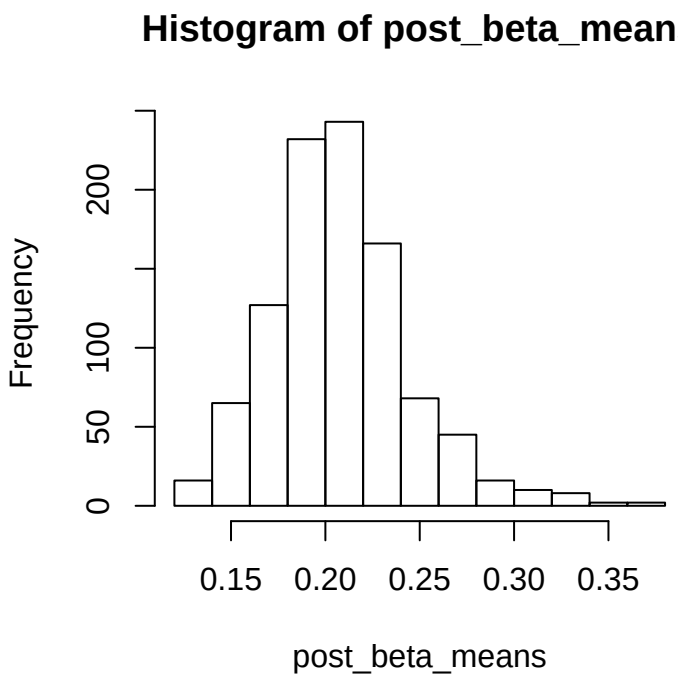
From this plot it is clear that the posterior estimates cluster around the observed proportions nicely. As we chose a non informative prior, this should not be surprising. If our prior distribution on (α, β) had a more concentrated shape then we would expect more aggressive pooling in this application. We do note however, for large values of θ_j such as θ_{10} the posterior is shrunk back towards the other data points as the red dot is not in the center of its distribution.

(d)

Recall that the underlying proportion of traffic that is bicycles was modeled using the Beta(α, β) distribution. Therefore, the average underlying proportion is given by $\frac{\alpha}{\alpha+\beta}$. Hence we can calculate this average with our posterior samples of α, β as follows.

```
#estimate of true underlying prop of traffic that is bicyles
post_beta_means <- ab_s[,1] / (ab_s[,1] + ab_s[,2])

#plot results
hist(post_beta_means)
```




```
#get mean
center <- mean(post_beta_means)

#get 95\% qunatile
quantile(post_beta_means, prob = c(.025, .975))
```

```
##      2.5%     97.5%
## 0.1436314 0.2954677
```

Therefore we see that that the posterior interval of the underlying proportion of traffic that is bicycles is roughly (.14, .29).

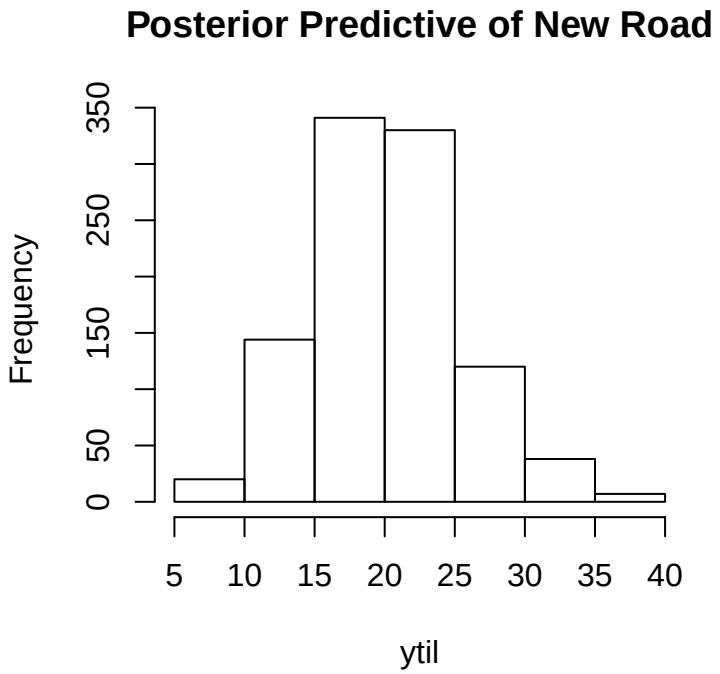
(e)

From, part (d) we estimated that the average underlying proportion of traffic that is bicycles is 0.2077244. Therefore, using this solution, we can draw samples from as

$$y|\hat{\theta}^{(b)} \sim \text{Binom}(100, \theta)$$

for each $b = 1, 2, \ldots, B$ to approximate the posterior distribution of this new road. This approach is implemented below.

```
#get posterior predictive samples
ytil <- sapply(post_beta_means, function(x) rbinom(1, 100, x))
hist(ytil, main = "Posterior Predictive of New Road")
```




```
#center
center <- mean(ytil)

#estimate 95% post interval
quantile(ytil, prob = c(.025, .975))
```

```
##  2.5% 97.5%
##    11    32
```

From here we see that the 95% posterior interval is roughly from $(11, 32)$ suggesting that of the 100 observes vehicles we expect to see 20.662 bicycles.

(f)

The Beta distribution could be reasonable but is more than likely not. The rate of which bicycles pass on a certain road more than likely is also dependent on certain covariates. Incorporating neighborhood information, weather information, as well as other covariates would more than likely be the most robust model in this setting.