PHP 2530 Bayesian Statistical Methods Homework I

Nick Lewis

2/17/2022

```
library(dplyr) #Allows us to use nice functions such as filter
library(ggplot2) #this makes better looking plots in R
library(latex2exp) #for using latex expressions in plots and names
```

Problem 3 (BDA 3rd. Ed. Exercise 1.9)

Simulation of a queuing problem: a clinic has three doctors. Patients come into the clinic at random, starting at 9 a.m., according to a Poisson process with time parameter 10 minutes: that is, the time after opening at which the first patient appears follows an exponential distribution with expectation 10 minutes and then, after each patient arrives, the waiting time until the next patient is independently exponentially distributed, also with expectation 10 minutes. When a patient arrives, he or she waits until a doctor is available. The amount of time spent by each doctor with each patient is a random variable, uniformly distributed between 15 and 20 minutes. The office stops admitting new patients at 4 p.m. and closes when the last patient is through with the doctor.

```
### PROBLEM 3 (BDA 3rd. Ed. 1.9)

poisson.process <- function(theta,time,a,b,num) {
  "
  PARAMETERS
   lambda - rate
   time - time period we're interested in (lambda and time must be same scale)
   a, b - time interval of time spent with patient. i.e. U ~ uniform(a,b)
   num - number of doctors
  "
  #samples 10*mean(Poisson(lambda*t)) from T ~ Exp(lambda) and sums them.
  #Removes those which exceed the time period
  arr.T <- cumsum(rexp(n=10*time/theta,rate=1/theta)) %>% .[.<= time]
  # records appointment duration wrt opening time
  doc <- rep(0,num)
  wait <- c()
  for(j in 1:length(arr.T) ) {
    # waiting time of patient j
    wait <- c(wait,min(doc) - arr.T[j])
    #appointment duration(if wait>0, appointment starts when doc finishes)
    doc[which.min(doc)] <-ifelse(wait[j]>=0,min(doc),arr.T[j])+runif(1,a,b)
  }
  #if wait <= 0, they didn't wait. If wait > 0, they did
  number.waited <- sum(ifelse(wait > 0, 1, 0))
  #waiting time is simply sum of positive waiting times
  time.waiting <- sum(wait[wait > 0])
  #time when office closes
  closing.time <- max(max(doc),time)
  average.wait.time <- ifelse(number.waited==0,0,time.waiting / number.waited)

  ### STORES OUR INFORMATION
  info <- matrix(c(length(arr.T), number.waited,
                   average.wait.time,  closing.time),nrow=1)
  colnames(info) <- c("Number of Arrivals","Number of Patients who Waited",
                      " Average Waiting Time", "Closing Time")
  return(info)
}
```

(a) Simulate this process once. How many patients came to the office? How many had to wait for a doctor? What was their average wait? When did the office close?

Solution:

```
#PART A

poisson.process(theta=10,time=420,a=15,b=20,num=3)
```

```
##      Number of Arrivals Number of Patients who Waited  Average Waiting Time
## [1,]                 51                            20              12.55733
##      Closing Time
## [1,]     430.0136
```

(b) Simulate the process 100 times and estimate the median and 50% interval for each of the summaries in (a).

Solution:

```
#get 100 samples from the Poisson process.
#use replicate instead of sapply since it keeps the names of the variables
samples <- replicate(100,poisson.process(theta=10,time=420,a=15,b=20,num=3))

apply(samples,2, quantile, probs=c(0.25, .50, .75))
```

```
##     Number of Arrivals Number of Patients who Waited  Average Waiting Time
## 25%              37.00                             7              4.498570
## 50%              41.00                            10              6.060707
## 75%              46.25                            17              8.165346
##     Closing Time
## 25%     424.9263
## 50%     430.6408
## 75%     435.6687
```

Problem 4: BDA 3rd Ed. Exercise 2.4

Predictive distributions: let y be the number of 6's in 1000 independent rolls of a particular real die, which may be unfair. Let θ be the probability that the die lands on '6.' Suppose your prior distribution for θ is as follows:

$$Pr(\theta = 1/12) = 0.25$$
$$Pr(\theta = 1/6) = 0.50$$
$$Pr(\theta = 1/4) = 0.25$$

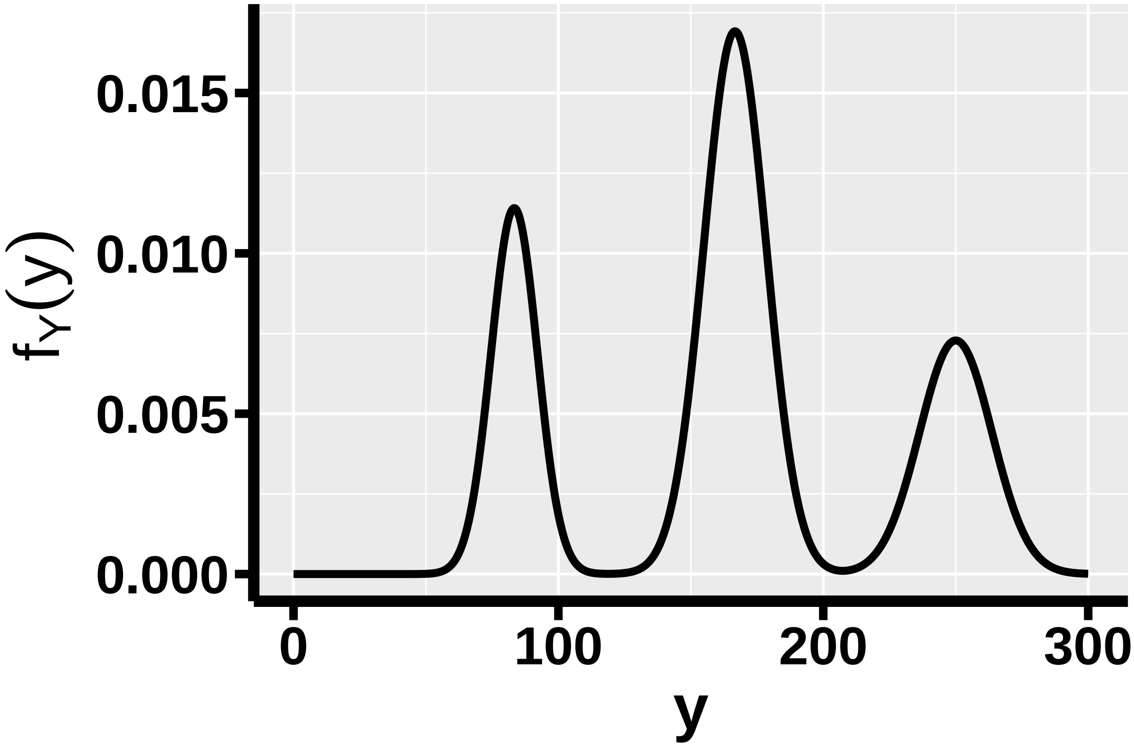
(a) Using the normal approximation for the conditional distributions, $p(y|\theta)$, sketch your approximate prior predictive distribution for y.

Solution:

```
### PROBLEM 4 (BDA 3rd. Ed. Exercise 2.4)
y <- seq(0, 300, length=1000)
fy <- function(x, theta){
  dnorm(x, mean = 1000*theta, sd = sqrt(1000*theta*(1-theta)))
}

#calculates fy for each theta giving 1000x3 matrix.
#Then matrix multiplication to give 1000 length vector i.e. (1000x3)(3x1)
p <- outer(y,c(1/12,1/6,1/4),fy) %*% c(0.25,0.5,0.25)

data <- data.frame(y, p)
ggplot(data = data, aes(y, p)) +
  geom_line(color="black",size=1.4)+
  theme(axis.line = element_line(colour = "black",size=2),
        text = element_text(size=20),
        axis.text = element_text(colour = "black",size = 20,face="bold"),
        axis.title = element_text(size = 24,face="bold"),
        axis.ticks.length=unit(.25, "cm"),
        axis.ticks = element_line(colour = "black", size = 1.5))+
  ylab(~ paste(f[Y](y)))+
  xlab("y")
```



(b) Give approximate 5%, 25%, 50%, 75%, and 95% points for the distribution of y. (Be careful here: y does not have a normal distribution, but you can still use the normal distribution as part of your analysis.)

Solution:

One strategy to find the quantiles is to recognize that the weights (i.e. the prior probability values) reveal how much each individual gaussian contributes to the pdf.

The leftmost Gaussian contributes 25% of the mass, so the 5% quantile is simply the 20% quantile of this Gaussian (i.e. 20% of 25 is 5).

Following similar logic the 25% quantile is directly between the first spike and the second spike (through trial and error I have found that taking the 99.997% of the first Gaussian gives extremely close results).

By symmetry, the 50% quantile is directly in the middle of the second spike.

The 75% quantile is directly between the second and third spikes (taking the 99.96% quantile of the second provides accurate results),

The 95% quantile is given by the 80% quantile of the third spike since the previous two gaussians contribute 75% of the mass, we look for where the third gaussian contributes 20% (i.e. 80% of 25 is 20).

```
#METHOD 1:

#Note, use 0.9999 for the quantiles in between. 1 gives you Infinity
q <- c(0.20,0.99997,0.50,0.9996,0.80)
mu.theta <- c(1/12,1/12,1/6,1/6,1/4)
sd.theta <- sqrt(1000*mu.theta*(1-mu.theta)); mu.theta <- 1000*mu.theta

Q <- sapply(1:length(q),function(x) qnorm(q[x],mu.theta[x],sd.theta[x]))
```

```
Q <- matrix(round(Q,3),nrow=1)
colnames(Q) <- c("5%","25%","50%","75%","95%")
print(Q)
```

```
##           5%     25%     50%    75%     95%
## [1,] 75.978 118.406 166.667 206.18 261.524
```

Another equally valid method is to recognize that

$$\int_{-\infty}^{y} p(y')dy' = \int_{-\infty}^{y} \sum_{\theta} p(y'|\theta)p(\theta)dy' = \sum_{\theta} \int_{-\infty}^{y} p(y'|\theta)p(\theta)dy'$$
$$F(y) = \sum_{\theta} F(y|\theta)p(\theta)$$

This gives us a form for the cdf of a mixture model. From here we can use a line-search method to find the values y such that $F(y) - q = 0$, where q is our quantile value.

```
#METHOD 2:

# GMQ- Gaussian Mixture Quantiles
GMQ <- function(p,theta,w,y){
  '
    Parameters
    ----------
    p : quantiles we wish to obtian values for
    theta : finite parameter space for theta
    w : weights attached to each theta
    y : upper bound of range to search over (i.e. we look from [0,y])

    Returns
    -------
    Quantiles of the gaussian mixture model

  '

  #functions to use
  gmm <- function(x, theta) {
    pnorm(x, mean = 1000*theta, sd = sqrt(1000*theta*(1-theta)))
    }
  N <- length(p)
  #initialize range to search over
  X <- seq(from=0,to=y,by=0.01);
  G <- (outer(X,theta,gmm) %*% w)
  #finding position of minimum value, then finding
  quantiles <- sapply(1:N,function(x) X[which.min(abs(G - p[x]))])
  quantile.names <-  sapply(1:N,function(x) paste0(100*p[x],"%"))
  #Nice, readable form
  quantiles <- matrix(quantiles,nrow=1); colnames(quantiles) <- quantile.names
  return( quantiles)
}
#quantile values
GMQ(p=c(0.05,0.25,0.50,0.75,0.95),theta=c(1/12,1/6,1/4),w=c(1/4,1/2,1/4),y=500)
```

```
##          5% 25%    50%    75%    95%
## [1,] 75.98 118 166.67 206.45 261.52
```

Problem 6: BDA 3rd Ed. Exercise 2.8

Normal distribution with unknown mean: a random sample of n students is drawn from a large population, and their weights are measured. The average weight of the n sampled students is $\bar{y} = 150$ pounds. Assume the weights in the population are normally distributed with unknown mean θ and known standard deviation 20 pounds. Suppose your prior distribution for θ is normal with mean 180 and standard deviation 40.

```
### PROBLEM 6 (BDA 3rd. Ed. Exercise 2.8)
var.n <- function(n,a,b){ 1/(1/(a)^2 + n/(b)^2)  }
mu.n <- function(n,a,b,m,y){ var.n(n,a,b)*(m/(a)^2 + (n*y)/(b)^2)}

mu <- mu.n(c(10,10,100,100),40,20,180,150)
var <- var.n(c(10,10,100,100),40,20) + c(0,20^2,0,20^2)

stats <- sapply(c(0.025,0.975),function(x) qnorm(x,mean=mu , sd =sqrt(var ) ) )
stats <- round(stats,2)
```

(c) For $n = 10$, give a 95% posterior interval for θ and a 95% posterior predictive interval for $\tilde{y}$.

Solution:

```
#PART C
sprintf("The Posterior Interval (c) is [%s]",
        paste0(stats[1,], collapse = ', '))
```

```
## [1] "The Posterior Interval (c) is [138.49, 162.98]"
```

```
sprintf("The Posterior Predictive Interval for (c) is [%s]",
        paste0(stats[2,], collapse = ', '))
```

```
## [1] "The Posterior Predictive Interval for (c) is [109.66, 191.8]"
```

(d) Do the same for n = 100.

Solution:

```
# PART D
sprintf("The Posterior Interval (d) is [%s]",paste0(stats[3,], collapse = ', '))
```

```
## [1] "The Posterior Interval (d) is [146.16, 153.99]"
```

```
sprintf("The Posterior Predictive Interval for (d) is [%s]",
        paste0(stats[4,], collapse = ', '))
```

```
## [1] "The Posterior Predictive Interval for (d) is [110.68, 189.47]"
```

Problem 7: BDA 3rd Ed. Exercise 2.10

Discrete sample spaces: suppose there are N cable cars in San Francisco, numbered sequentially from 1 to N. You see a cable car at random; it is numbered 203. You wish to estimate N. (See Goodman, 1952, for a discussion and references to several versions of this problem, and Jeffreys, 1961, Lee, 1989, and Jaynes, 2003, for Bayesian treatments.)

(a) Assume your prior distribution on N is geometric with mean 100; that is,

$$p(N) = (1/100)(99/100)^{N-1}, \text{ for } N = 1, 2, \ldots$$

What is your posterior distribution for N?

(b) What are the posterior mean and standard deviation of N? (Sum the infinite series analytically or approximate them on the computer.)

Solution:

```
### PROBLEM 7 (BDA 3rd. Ed. Exercise 2.10)

#values to sum over
values <- c(203:10000)
#p(X)
p.X <- sum((1/100)*(1/values)*(99/100)^(values - 1))
sprintf("The normalizing constant for the posterior is %.7s",p.X)
```

```
## [1] "The normalizing constant for the posterior is 0.00047"
```

```
#p(N|X)
post <- (1/(100*p.X*values))*(99/100)^(values - 1)
#E(N|X)
mu.N <-  sum(values*post)
sprintf("The posterior mean is %.6s",mu.N)
```

```
## [1] "The posterior mean is 279.08"
```

```
#Var(N|X) = sum (N-E(N|X))^2 p(N|X)
sd.N <- sqrt(sum((values-mu.N)^2*post))
sprintf("The posterior standard deviation is %.5s",sd.N)
```

```
## [1] "The posterior standard deviation is 79.96"
```

(c) Choose a reasonable 'noninformative' prior distribution for N and give the resulting posterior distribution, mean, and standard deviation for N.

Solution:

```
# Part c (Poisson Prior)

#q(N|X), unnormalized posterior
#put everything in terms of log and exponents so R can handle computation
unnorm.post <- exp((values)*log(100)-lfactorial(values)-100-log(values))
#p(X)
p.X1 <- sum(unnorm.post)
#p(N|X)
new.post <- unnorm.post/p.X1
#E(N|X)
mu.N1 <-  sum(values*new.post)
sprintf("The posterior mean is %.6s",mu.N1)
```

```
## [1] "The posterior mean is 203.93"
```

```
#sd(N|X)
sd.N1 <- sqrt(sum((values-mu.N1)^2*new.post))
sprintf("The posterior standard deviation is %.5s",sd.N1)
```

```
## [1] "The posterior standard deviation is 1.334"
```

Problem 8: BDA 3rd Ed. Exercise 2.13

Discrete data: The table below gives the number of fatal accidents and deaths on scheduled airline flights per year over a ten-year period. We use these data as a numerical example for fitting discrete data models.

(a) Assume that the numbers of fatal accidents in each year are independent with a Poisson(θ) distribution. Set a prior distribution for θ and determine the posterior distribution based on the data from 1976 through 1985. Under this model, give a 95% predictive interval for the number of fatal accidents in 1986. You can use the normal approximation to the gamma and Poisson or compute using simulation.

```
### PROBLEM 8 (BDA 3rd Ed. Exercise  2.13)
#data for the problem
df <- data.frame(year =1:10 ,
                 accidents=c(24, 25, 31, 31, 22, 21, 26, 20, 16, 22),
                 deaths=c(734, 516, 754, 877, 814, 362, 764, 809, 223, 1066),
```

```r
                 rate =c(0.19, 0.12, 0.15, 0.16, 0.14, 0.06, 0.13, 0.13, 0.03, 0.15)
)
df["miles"] <- df["deaths"]*(1e8)/df["rate"]

#Prior distribution parameters (here so you can adjust for different priors)
prior.shape <- 0; prior.rate <- 0

## APPROACH 1: FIND POSTERIOR PREDICTIVE DISTRIBUTION

sizes <- c(sum(df["accidents"]),sum(df["accidents"]),
           sum(df["deaths"]),sum(df["deaths"])) + prior.shape

#corresponding probability parameters
probs <- c(nrow(df)/(nrow(df)+1+prior.rate),
               sum(df["miles"])/(sum(df["miles"])+(8e11)+prior.rate),
               nrow(df)/(nrow(df)+1+prior.rate),
               sum(df["miles"])/(sum(df["miles"])+(8e11)+prior.rate))

app1 <- sapply(c(0.025,0.975),function(x) qnbinom(x,size = sizes,prob= probs ) )

## APPROACH 2: SAMPLE FROM POSTERIOR, PLUG BACK INTO LIKELIHOOD
#Strategy:
#(sample from theta|y, plug values into y|theta, sort from least to greatest)
#find 25th and 975th values, these represent endpoints of 95% posterior interval

#shape, rate and miles
a <- c(sum(df["accidents"]),sum(df["accidents"]),
      sum(df["deaths"]),sum(df["deaths"])) + prior.shape

b <- c(nrow(df),sum(df["miles"]),nrow(df),sum(df["miles"])) + prior.rate

m <- c(1,8e11,1,8e11)

#shapes
app2 <- sapply(1:4,function(x) sort(rpois(1000,m[x]*rgamma(1000,a[x],b[x])))[c(25,975)])
app2 <- t(app2)
```

```r
# PART A

sprintf("The Posterior Interval for (a) using Method 1 is [%s]",
        paste0(app1[1,], collapse = ', '))
```

```
## [1] "The Posterior Interval for (a) using Method 1 is [14, 34]"
```

```r
sprintf("The Posterior Interval for (a) using Method 2 is [%s]",
        paste0(app2[1,], collapse = ', '))
```

```
## [1] "The Posterior Interval for (a) using Method 2 is [14, 34]"
```

(b) Assume that the numbers of fatal accidents in each year follow independent Poisson distributions with a constant rate and an exposure in each year proportional to the number of passenger miles flown.

Set a prior distribution for θ and determine the posterior distribution based on the data for 1976-1985. (Estimate the number of passenger miles flown in each year by dividing the appropriate columns of Table 2.2 and ignoring round-off errors.) Give a 95% predictive interval for the number of fatal accidents in 1986 under the assumption that 8×10^{11} passenger miles are flown that year.

Solution:

```
# PART B
sprintf("The Posterior Interval for (a) using Method 1 is [%s]",
        paste0(app1[2,], collapse = ', '))
```

```
## [1] "The Posterior Interval for (a) using Method 1 is [22, 46]"
```

```
sprintf("The Posterior Interval for (a) using Method 2 is [%s]",
        paste0(app2[2,], collapse = ', '))
```

```
## [1] "The Posterior Interval for (a) using Method 2 is [22, 46]"
```

(c) Repeat (a) above, replacing ‘fatal accidents’ with ‘passenger deaths.’

Solution:

```
# PART C
sprintf("The Posterior Interval for (a) using Method 1 is [%s]",
        paste0(app1[3,], collapse = ', '))
```

```
## [1] "The Posterior Interval for (a) using Method 1 is [638, 747]"
```

```
sprintf("The Posterior Interval for (a) using Method 2 is [%s]",
        paste0(app2[3,], collapse = ', '))
```

```
## [1] "The Posterior Interval for (a) using Method 2 is [640, 745]"
```

(d) Repeat (b) above, replacing ‘fatal accidents’ with ‘passenger deaths.’

Solution:

```
# PART D
sprintf("The Posterior Interval for (a) using Method 1 is [%s]",
        paste0(app1[4,], collapse = ', '))
```

```
## [1] "The Posterior Interval for (a) using Method 1 is [904, 1034]"
```

```
sprintf("The Posterior Interval for (a) using Method 2 is [%s]",
        paste0(app2[4,], collapse = ', '))
```

```
## [1] "The Posterior Interval for (a) using Method 2 is [902, 1034]"
```