Solution to computer exam in Bayesian learning

Per Siden

2020-06-04

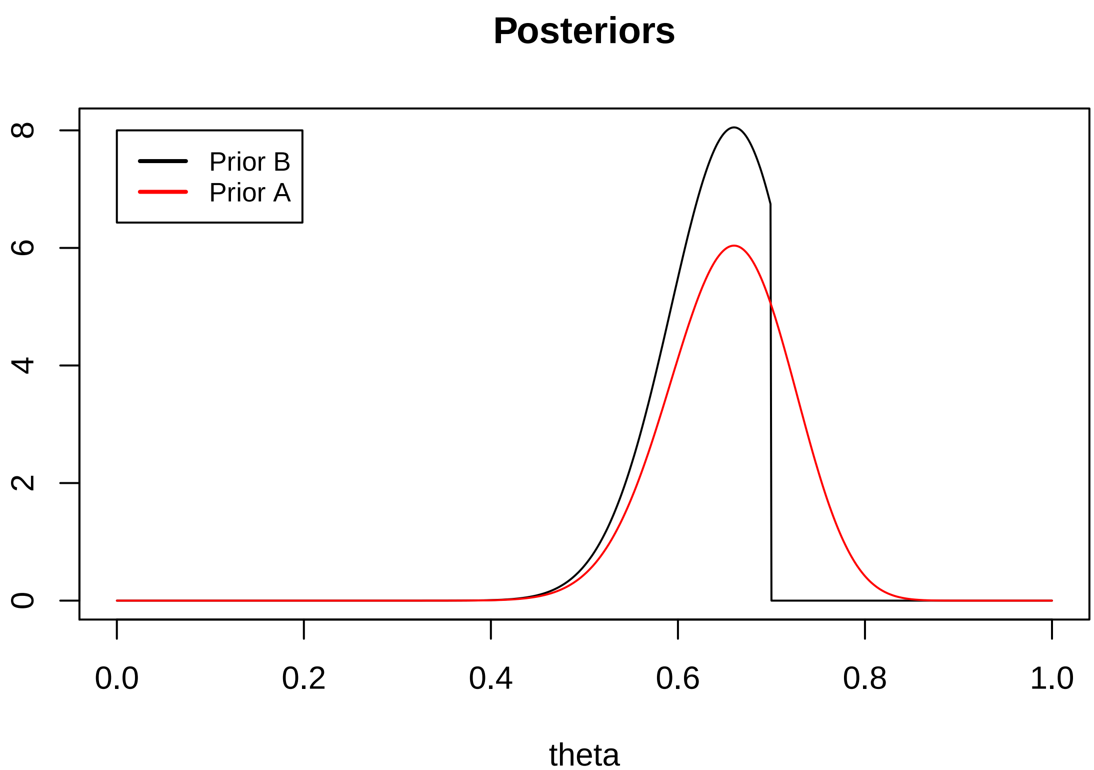
First load all the data into memory by running the R-file given at the exam

```
rm(list=ls())
source("ExamData.R")
set.seed(1)
```

Problem 1

1b

```
x=33
gridstep <- 0.001
thetaGrid <- seq(0,1,gridstep)
unnormpost <- dbinom(x,50,thetaGrid)*(thetaGrid>0.3)*(thetaGrid<0.7)
postB <- 1/gridstep*unnormpost/sum(unnormpost)
plot(thetaGrid,postB,type="l",main="Posteriors",xlab="theta",ylab="")
lines(thetaGrid,dbeta(thetaGrid,x+1,51-x),type="l",col=2)
legend(x = 0, y = 8, legend = c("Prior B","Prior A"),
       col = c("black","red"), lty = c(1,1), lwd = c(2,2), cex = 0.8)
```



1c

```
ProbA <- pbeta(0.5,x+1,51-x)
ProbB <- sum(postB[thetaGrid<=0.5]*gridstep)
print(ProbA)
```

```
## [1] 0.01204645
```

```r
print(ProbB)
```

```
## [1] 0.0163567
```

The probability is 0.012 under prior A and 0.016 under prior B.

Problem 2

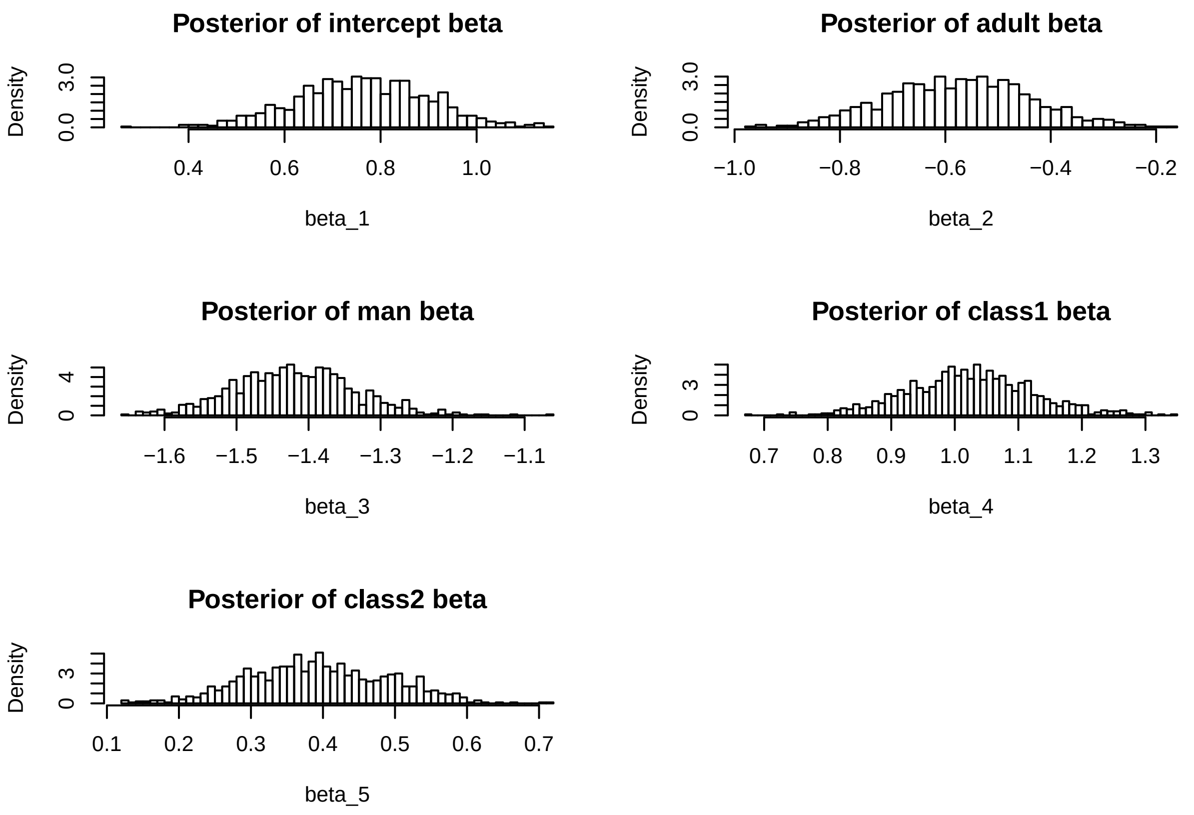
2a

```r
library(mvtnorm)

y <- as.vector(titanic[,1])
X <- as.matrix(titanic[,-1])
covNames <- names(titanic)[2:length(names(titanic))]
nPara <- dim(X)[2]

# Setting up the prior
tau <- 50
mu <- as.vector(rep(0,nPara)) # Prior mean vector

nIter = 1000
betaSample <- BayesProbReg(y, X, mu, tau, nIter)
par(mfrow=c(3,2))
for(i in 1:5){
  hist(betaSample[,i], 50, freq=FALSE,
       main = paste('Posterior of',covNames[i],'beta'), xlab = paste0('beta_',i))
}
```



2b

The posterior median is the optimal estimator under linear loss function (see Slides Lecture 4)

```
medians <- apply(betaSample,2,median)
names(medians) <- covNames
print(medians)
```

```
##   intercept      adult        man     class1     class2
##   0.7624990 -0.5723066 -1.4225862  1.0229195  0.3907436
```

2c

```
postProb = mean(betaSample[,2]+betaSample[,5]>0)
print(postProb)
```

```
## [1] 0.144
```

The probability is roughly 0.13 which is the probability that an adult travelling in 2nd class is more likely to survive than another type of passenger.

Problem 3

3a

```
y <- matrix(c(5,3,17,8),4,1)
x <- matrix(log(c(20,20,50,40)),4,1)

library("mvtnorm")
LogPostPoisson <- function(beta,y,x){
  linPred <- x*beta
  logLik <- sum(dpois(y, exp(linPred), log = TRUE))
  if (abs(logLik) == Inf) logLik = -20000;
  logPrior <- dnorm(beta, 1, 0.1, log=TRUE);
  # logPrior <- dnorm(beta, 0,100, log=TRUE);
  return(logLik + logPrior)
}

initVal <- 0

OptimResults<-optim(initVal,LogPostPoisson,gr=NULL,y,x,method=c("BFGS"),
                    control=list(fnscale=-1),hessian=TRUE)
postMode <- OptimResults$par
postStd <- sqrt(-solve(OptimResults$hessian))
mean <- round(postMode,digits=2)
std <- round(postStd,digits=2)
print(paste("Mean: ",mean))
```

```
## [1] "Mean:  0.69"
```

```
print(paste("Sd: ",std))
```

```
## [1] "Sd:  0.04"
```

The posterior mean and standard deviation are reported above.

3b

Simulate from the predictive posterior of y_5 by first sampling from the posterior of β and then from the likelihood given the two values of x.

```
loss <- function(y,x){
  return(4+0.02*exp(x)-sqrt(y))
}
expectedLoss <- function(xtest,nSamples){
  betaSim <- rnorm(nSamples,postMode,postStd)
  linPred <- xtest*betaSim
  ySim <- rpois(nSamples,exp(linPred))
  return(mean(sapply(ySim,loss,x=xtest)))
}
nSamples = 10000
xgrid = log(c(20,40))
EL = c()
for(i in 1:length(xgrid)){
  EL[i] <- expectedLoss(xgrid[i],nSamples)
}
print(paste('Spending',exp(xgrid[1]),'leads to expected loss:',round(EL[1],2)))
```

```
## [1] "Spending 20 leads to expected loss: 1.64"
```

```
print(paste('Spending',exp(xgrid[2]),'leads to expected loss:',round(EL[2],2)))
```

```
## [1] "Spending 40 leads to expected loss: 1.28"
```

The computed expected loss is lower when spending 40 million, so this is how much the country should spend.

Problem 4

4c

```
xbar_FL = 14
xbar_FW = 300
xbar_ML = 12
xbar_MW = 280
sigma_L = 2
sigma_W = 50

prob_unnorm_F = dnorm(10,14,sigma_L*sqrt(1+1/16))*dnorm(250,300,sigma_W*sqrt(1+1/16))*.75
prob_unnorm_M = dnorm(10,12,sigma_L*sqrt(1+1/4))*dnorm(250,280,sigma_W*sqrt(1+1/4))*.25
prob_F = prob_unnorm_F/(prob_unnorm_F+prob_unnorm_M)
print(prob_F)
```

```
## [1] 0.3663727
```

The predicitive probability is 0.37.