Solution to computer exam in Bayesian learning

Per Siden

2020-08-19

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

1a

```
N <- 100000
thetaSim <- rnorm(N,10000,500)
# print(quantile(thetaSim,c(0.05,0.95))) # Simulation
print(qnorm(c(0.05,0.95),10000,500)) # Exact
```

```
## [1]  9177.573 10822.427
```

The 90% credible interval is (9180,10820).

1b

```
xSim <- rpois(N,thetaSim)
print(quantile(xSim,c(0.05,0.95)))
```

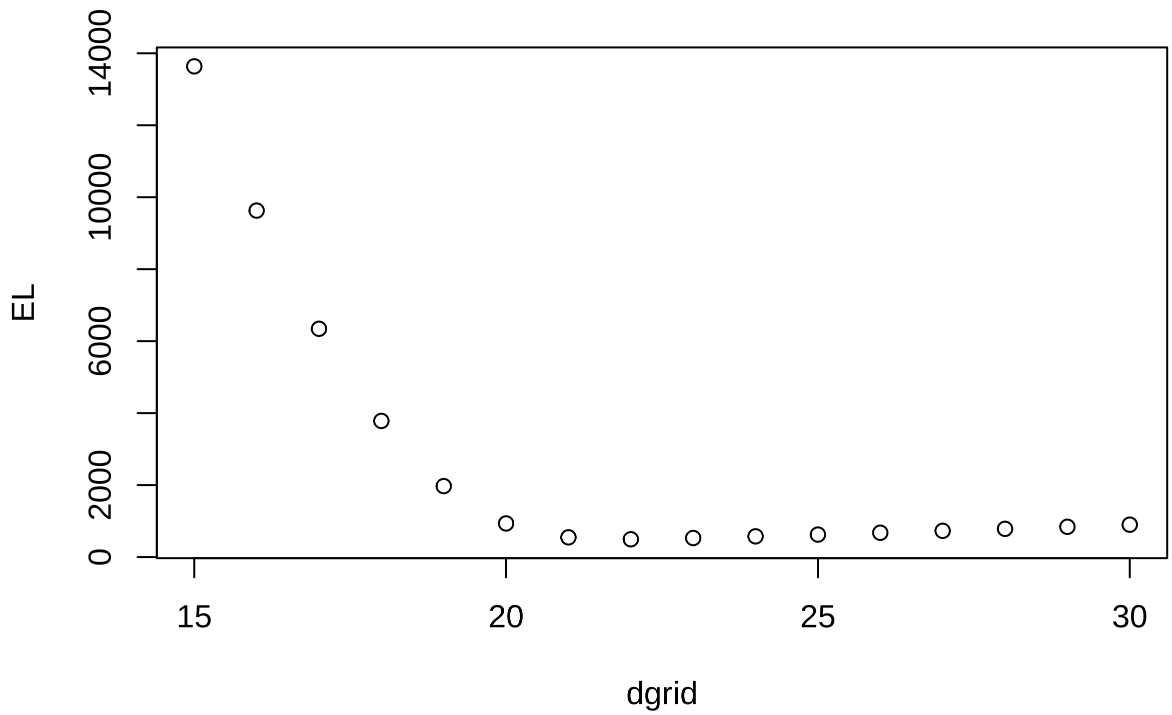
```
##    5%   95%
##  9160 10842
```

The 90% prediction interval is roughly (9160,10840).

1c

```
L <- function(p,d){
  return((d+0.04*pmax(p-500*d,0))^2)
}
dgrid <- seq(15,30,1)
EL <- 0*dgrid
count <- 1

for(d in dgrid) {
  EL[count] <- mean(L(xSim,d))
  count <- count + 1
}
plot(dgrid,EL)
```




```
print(dgrid[which.min(EL)])
```

```
## [1] 22
```

Given that the expected loss is minimized at d=22, the health centre should hire two more doctors.

Problem 2

2a

```
y = muscle$Length
X1 = as.matrix(muscle$Conc)
X2 = cbind(rep(1,length(muscle$Conc)),muscle$Conc,muscle$Conc^2)

RegFunc <- function(y,X){
  nCovs = dim(X)[2]
  mu_0 = rep(0,nCovs)
  Omega_0 = 0.001*diag(nCovs)
  v_0 = 1
  sigma2_0 = 10
  BayesLinReg(y, X, mu_0, Omega_0, v_0, sigma2_0, nIter = 5000)
}

M1 <- RegFunc(y,X1)
M2 <- RegFunc(y,X2)

M1_interval <- quantile(M1$betaSample,c(0.025,0.975))
M2_interval0 <- quantile(M2$betaSample[,1],c(0.025,0.975))
M2_interval1 <- quantile(M2$betaSample[,2],c(0.025,0.975))
M2_interval2 <- quantile(M2$betaSample[,3],c(0.025,0.975))
rbind(M1_interval,M2_interval0,M2_interval1,M2_interval2)
```

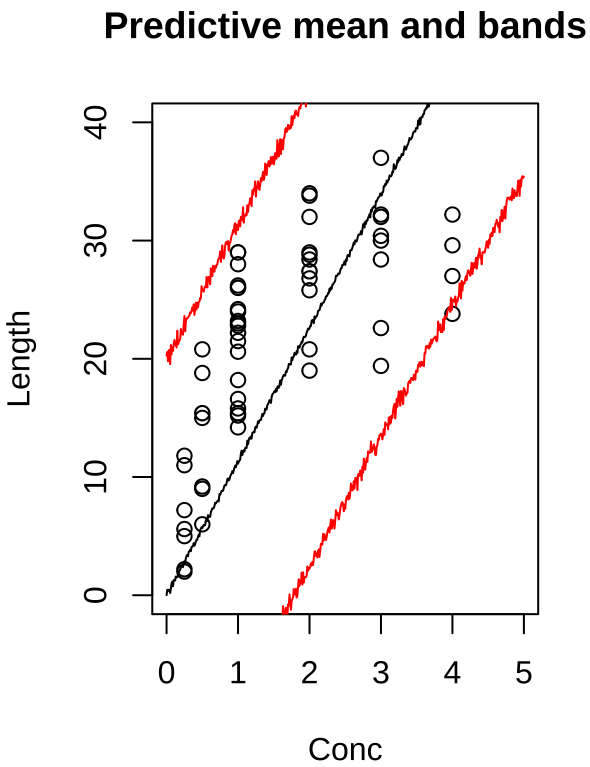
```
##                   2.5%     97.5%
## M1_interval    9.944439 12.669327
## M2_interval0   1.514350  8.581605
## M2_interval1  14.360279 23.352751
## M2_interval2  -4.494110 -2.323867
```

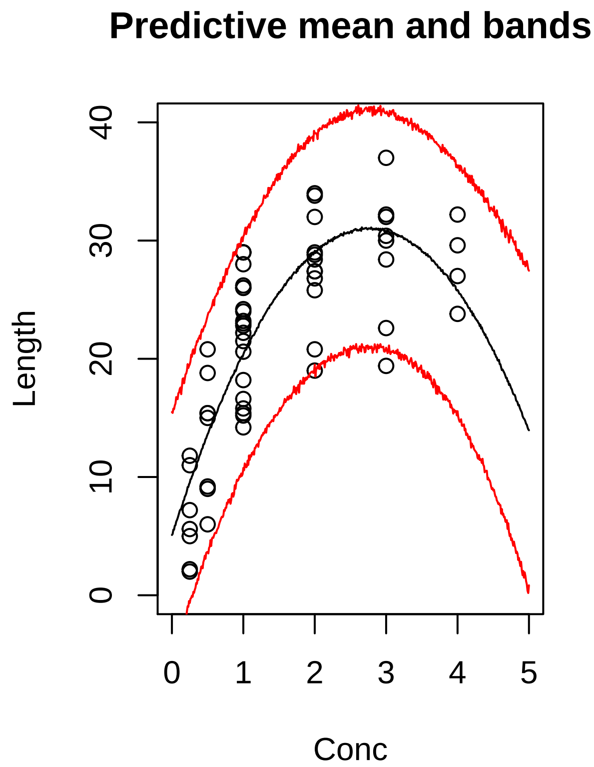
2b

```r
conc_grid = seq(0,5,0.01)
X1pred <- as.matrix(conc_grid)
X2pred <- cbind(rep(1,length(conc_grid)),conc_grid,conc_grid^2)
n = length(conc_grid)
PredDist <- function(X,M){
  predmean = matrix(0,n,1)
  predbands = matrix(0,n,2)
  for(i in 1:n){
    samps = X[i,]%*%t(M$betaSample) + rnorm(5000,0,sqrt(M$sigma2Sample))
    predmean[i] = mean(samps)
    predbands[i,] = quantile(samps,probs=c(.025,.975))
  }
  return(list(predmean = predmean, predbands=predbands))
}

pred1 <- PredDist(X1pred,M1)
pred2 <- PredDist(X2pred,M2)

par(mfrow=c(1,2))
plot(muscle$Conc,muscle$Length,type='p',xlim=c(0,5),ylim=c(0,40),
     xlab="Conc",ylab="Length",main="Predictive mean and bands")
lines(conc_grid,pred1$predmean)
lines(conc_grid,pred1$predbands[,1],col=2)
lines(conc_grid,pred1$predbands[,2],col=2)
plot(muscle$Conc,muscle$Length,type='p',xlim=c(0,5),ylim=c(0,40),
     xlab="Conc",ylab="Length",main="Predictive mean and bands")
lines(conc_grid,pred2$predmean)
lines(conc_grid,pred2$predbands[,1],col=2)
lines(conc_grid,pred2$predbands[,2],col=2)
```





Problem 3

3b

```r
n <- length(delivery)
N <- 1000
alpha <- 2
beta <- 2
ks <- c(0.5,1.5,2.5)
modes <- rep(0,3)
# exact_modes <- rep(0,3)
for(i in 1:3){
  inv_lambda <- rgamma(N,alpha+n,beta+sum(delivery^ks[i]))
  lambda <- 1/inv_lambda
  dens <- density(lambda)
  # plot(dens)
  # exact_modes[i] <- (beta+sum(delivery^ks[i]))/(1+alpha+n)
  modes[i] <- dens$x[which.max(dens$y)]
}
# print(exact_modes)
print(modes)
```

```
## [1]  1.581692  5.400066 27.241551
```

The posterior modes for lambda are roughly 1.6, 5.6 and 26.6 for the three k-values.

3c

```r
hist(delivery,breaks=30,freq=F,xlab="x")
xgrid <- seq(0,10,length.out=1000)
```

```r
# log_weibull_pdf <- function(x,k,lambda){
#   log_pdf <- log(k) - log(lambda) + (k-1)*log(x) -x^k/lambda
# }
weibull_pdf <- function(x,k,lambda){
  pdf <- k/lambda*x^(k-1)*exp(-x^k/lambda)
}
for(i in 1:3){
  # lines(xgrid,exp(log_weibull_pdf(xgrid,ks[i],modes[i])),col=i+1)
  lines(xgrid,weibull_pdf(xgrid,ks[i],modes[i]),col=i+1,)
}
legend(x=6,y=.25,c("k=0.5","k=1.5","k=2.5"),col=seq(2,4),lty="solid")
```

The Weibull density for $k = 1.5$ seems to give the best fit to the data. The density for $k = 0.5$ has too much mass for x-values close to 0 and for large values, and the density for $k = 2.5$ instead has too little mass there.

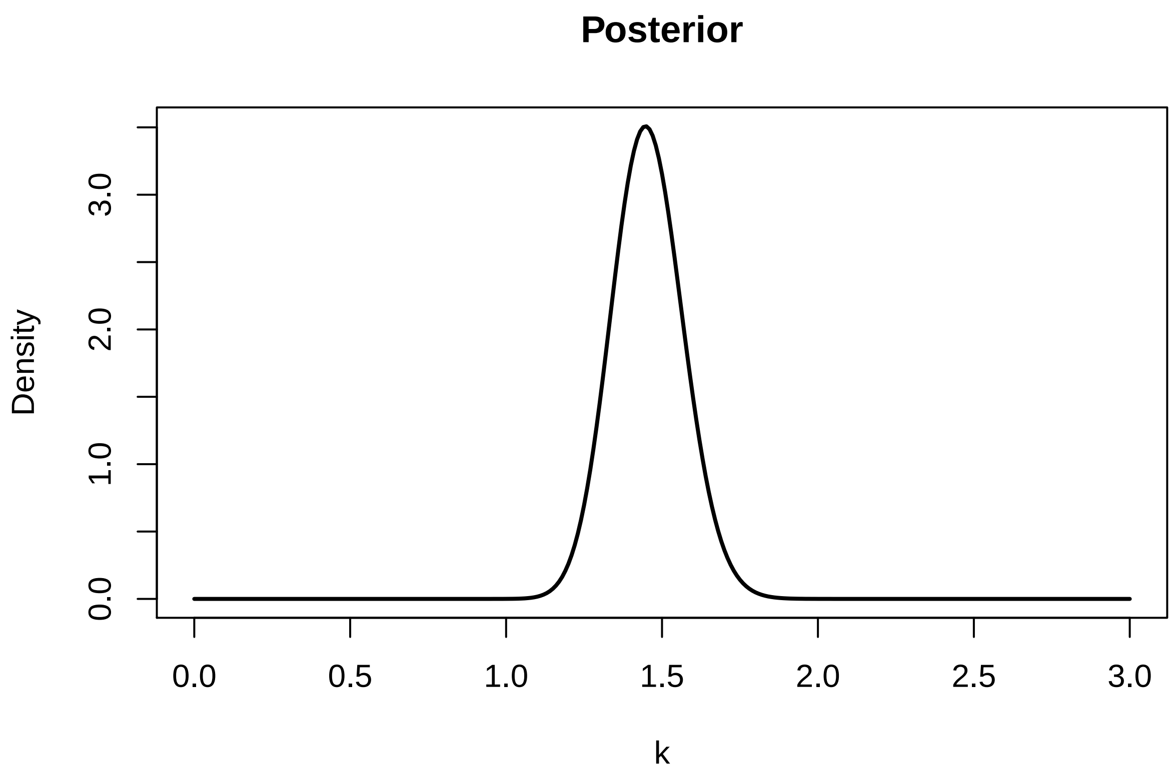
Problem 4

4b

```r
n <- length(delivery)
alpha <- 2
beta <- 2
log_like <- function(x,k){
  log_like <- lgamma(alpha+n) - lgamma(alpha) + alpha*log(beta) + n*log(k) +
              sum((k-1)*log(x)) - (alpha+n)*log(beta+sum(x^k))
}
log_posterior <- function(x,k){
  log_post <- log_like(x,k) + dexp(k,1,log=TRUE)
}
```

```r
gridWidth <- .01
kGrid <- seq(0, 3, by = gridWidth)
logPostGrid <- rep(0,length(kGrid))
count <- 0
for (k in kGrid){
  count <- count + 1
  logPostGrid[count] <- log_posterior(delivery,k)
}

# Taking exp() and normalizing so the density integrates to one. Note the (1/gridWidth) factor.
postGrid <- (1/gridWidth)*exp(logPostGrid)/sum(exp(logPostGrid))
plot(kGrid, postGrid, type = "l", lwd = 2,main="Posterior",
     ylab = "Density", xlab = "k")
```



4c

```r
kPostCDF <- cumsum(postGrid)*gridWidth
lowerBound <- kGrid[which.min(abs(kPostCDF-0.025))]
upperBound <- kGrid[which.min(abs(kPostCDF-0.975))]
print(c(lowerBound,upperBound))
```

```
## [1] 1.23 1.68
```

The 95% credible interval for k is (1.23,1.68).