Solution to computer exam in Bayesian learning

Bertil Wegmann

2021-10-21

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)
```

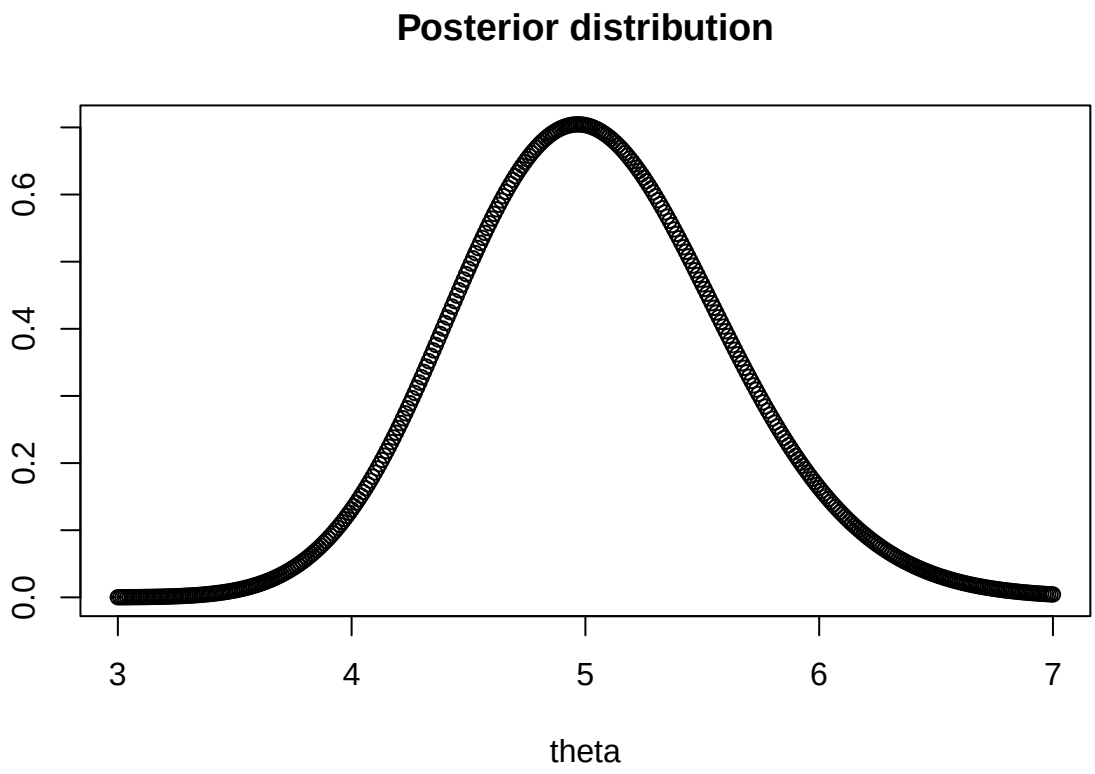
Problem 1

1d

```
LogPost <- function(theta,n,Sumx){

  logLik <- Sumx*log(theta) - n*theta;
  logPrior <- 2*log(theta) - 0.5*theta;

  return(logLik + logPrior)
}
theta_grid <- seq(3,7,0.01)
PostDens_propto <- exp(LogPost(theta_grid,15,75))
PostDens <- PostDens_propto/(0.01*sum(PostDens_propto))
plot(theta_grid,PostDens,main="Posterior distribution",xlab="theta", ylab="")
```

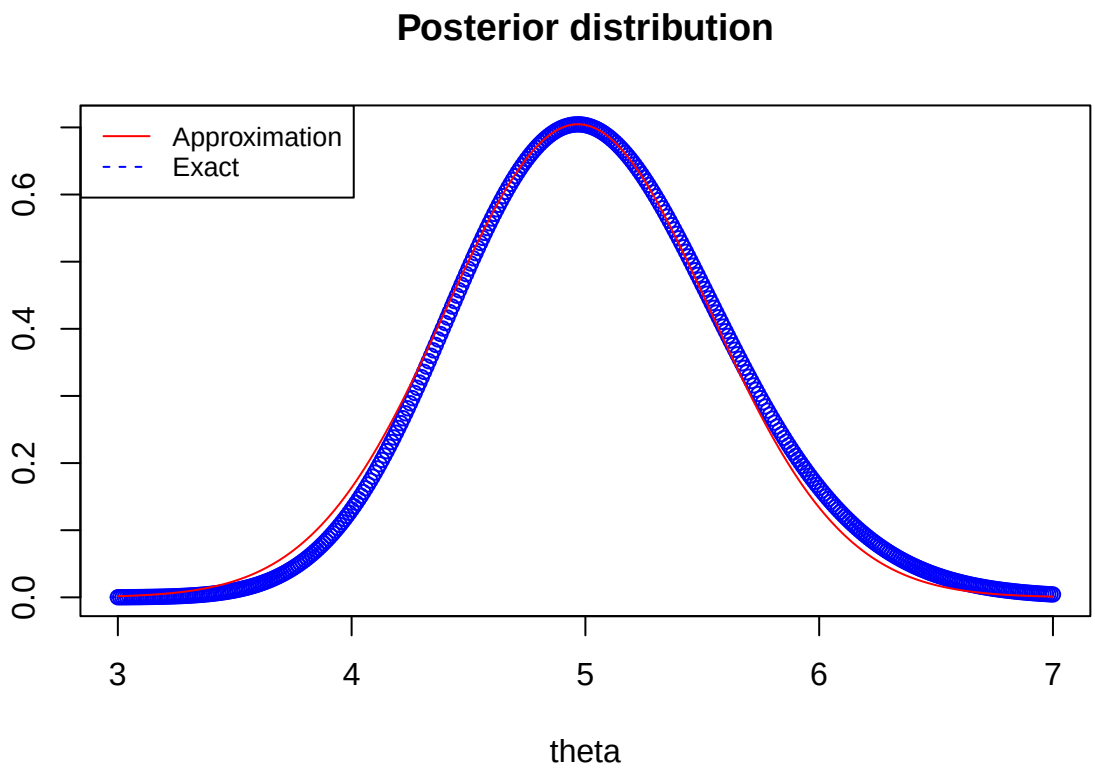


The posterior distribution is given above.

1e

```
n <- 15
Sumx <- 75
OptRes <- optim(4,LogPost,gr=NULL,n,Sumx,method=c("L-BFGS-B"),lower=3,
                control=list(fnscale=-1),hessian=TRUE)

plot(theta_grid,PostDens,col="blue",main="Posterior distribution",xlab="theta", ylab="")
lines(theta_grid,dnorm(theta_grid,mean = OptRes$par,sd = sqrt(-1/OptRes$hessian)),col="red")
legend("topleft", legend=c("Approximation", "Exact"), col=c("red", "blue"), lty=1:2, cex=0.8)
```



The posterior approximation is very accurate and the exact posterior distribution is only slightly skewed to the right.

1f

```
nSim <- 1e5
T_x_rep <- matrix(0,nSim,1)
for (ii in 1:nSim){
  theta <- rgamma(1,shape = 3+Sumx,rate = 0.5+n)
  x_rep <- rpois(n,theta)
  T_x_rep[ii,1] <- max(x_rep)
}
mean(T_x_rep >= 14)
```

```
## [1] 0.0164
```

The posterior predictive p-value is roughly 0.016, so the probability is very low that the maximum value of 14 from Gunnar originates from the current Poisson distribution.

Problem 2

2a

```
mu_0 <- as.vector(rep(0,3))
Sigma_0 <- 4**2*diag(3)
nIter <- 10000
library(mvtnorm)
```

```
X <- as.matrix(X)
PostDraws <- BayesLogitReg(y, X, mu_0, Sigma_0, nIter)
Betas <- PostDraws$betaSample
quantile(Betas[,2],probs=c(0.05,0.95))
```

```
##        5%       95%
## 0.2175566 1.8769813
```

It is 90 % posterior probability that beta_1 is on the interval (0.22,1.88).

2b

```
mean(Betas[,3]>0)
```

```
## [1] 0.8886
```

The posterior probability is roughly 0.89, i.e. the probability that x_2 has a positive effect on p_i when x_2 changes from 0 to 1.

2c

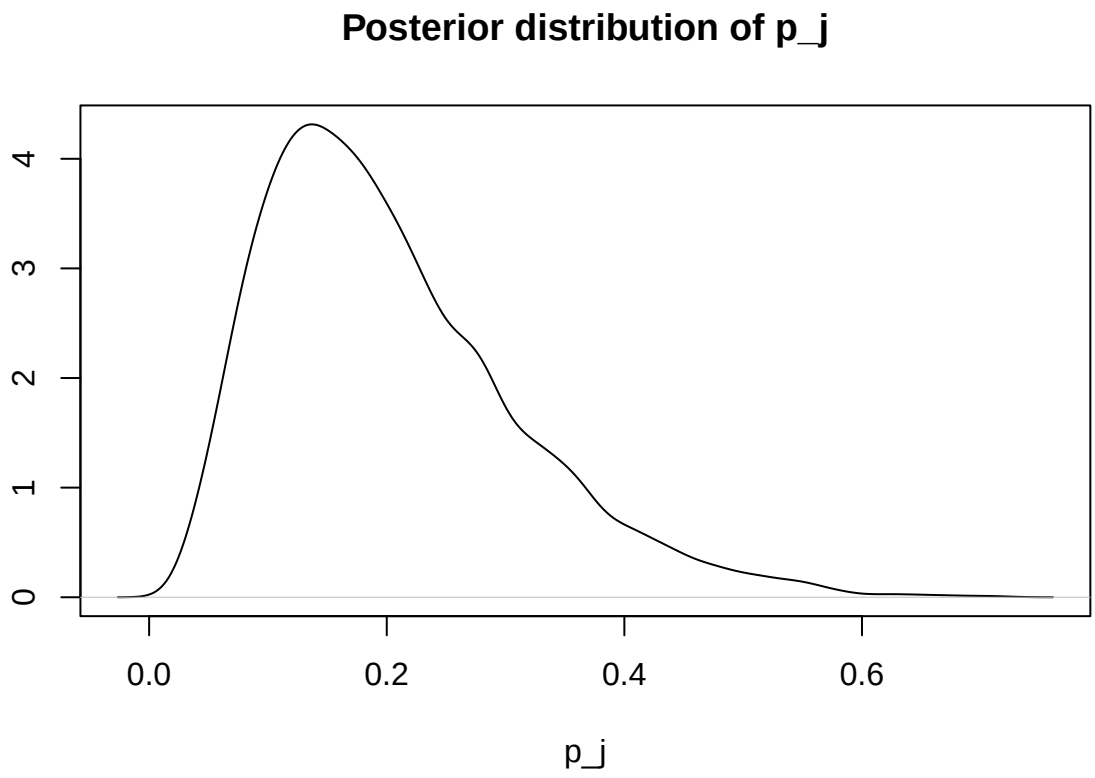
```
mean(Betas[,2]>0 & Betas[,3]>0)
```

```
## [1] 0.8773
```

The joint posterior probability that both beta_1 and beta_2 are positive is roughly 0.88.

2d

```
p_j_draws <- exp(Betas[,1])/(1+exp(Betas[,1]))
plot(density(p_j_draws),type="l",main="Posterior distribution of p_j",xlab="p_j",ylab="")
```




```
mean(p_j_draws>0.5)
```

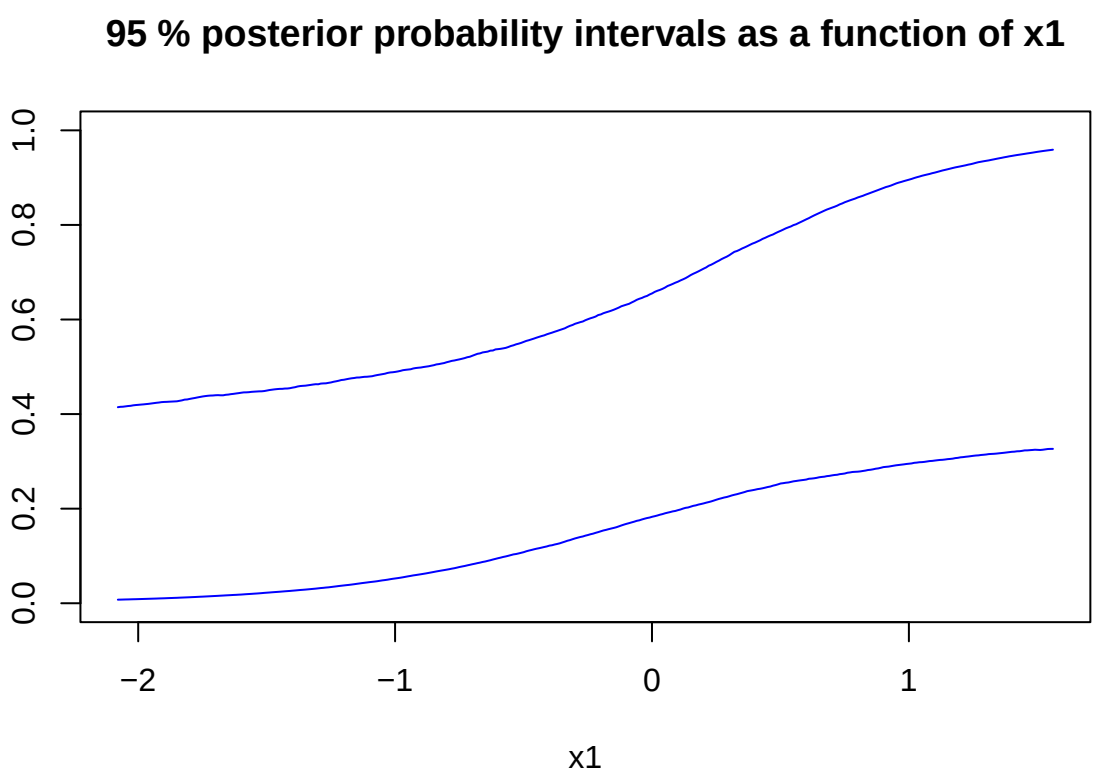
```
## [1] 0.0154
```

The posterior distribution of p_j is plotted above. The posterior probability is roughly 0.015.

2e

```
x1_grid <- seq(min(X[,2]),max(X[,2]),0.01)
p_k_draws <- matrix(0,length(x1_grid),2)
for (ii in 1:length(x1_grid)){
  LinPred <- Betas%*%as.vector(c(1,x1_grid[ii],1))
  Curr_p_k <- exp(LinPred)/(1+exp(LinPred))
  p_k_draws[ii,] <- quantile(Curr_p_k,probs=c(0.025,0.975))
}

plot(x1_grid,p_k_draws[,1],"n",
     main="95 % posterior probability intervals as a function of x1",
     xlab="x1", ylab="",ylim=c(0,1))
lines(x1_grid,p_k_draws[,1],col="blue")
lines(x1_grid,p_k_draws[,2],col="blue")
```

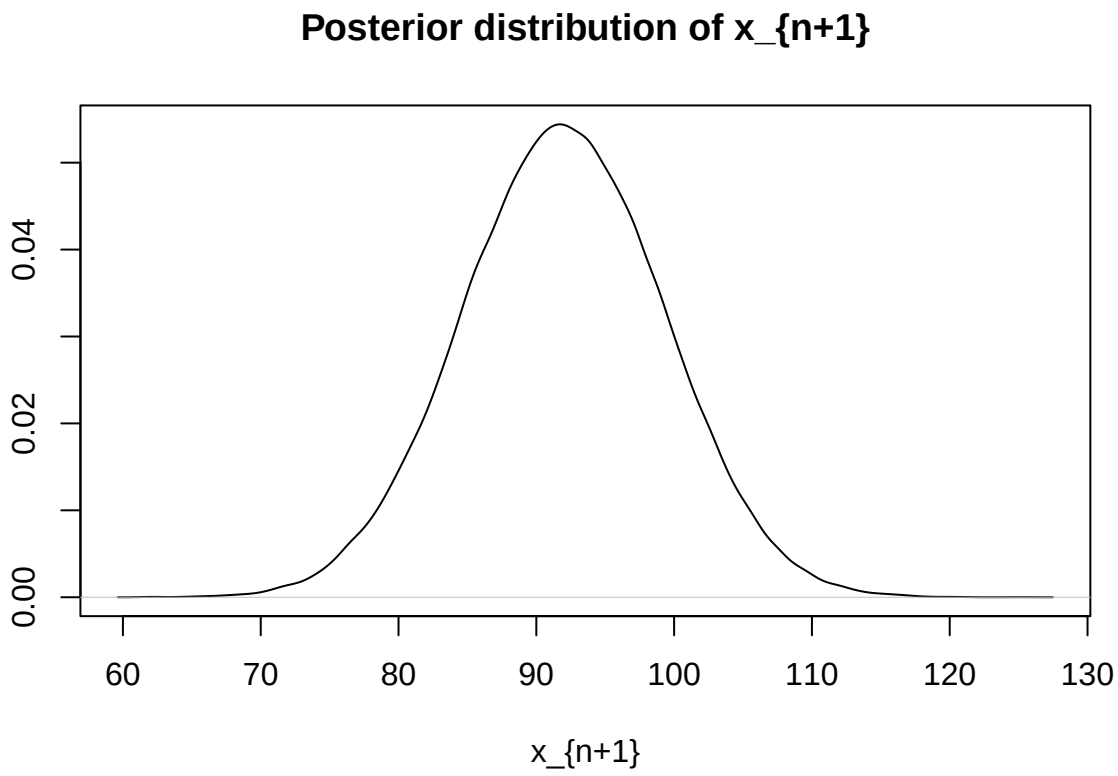


The posterior probability intervals as a function of x1 are plotted above.

Problem 3

3b

```
nSim <- 1e5
x_pred <- matrix(0,nSim,1)
for (jj in 1:nSim){
  mu_draw <- rnorm(1,mean = 92,sd = 2)
  x_pred[jj,1] <- rnorm(1,mean = mu_draw,sd = sqrt(50))
}
plot(density(x_pred),type="l",main="Posterior distribution of x_{n+1}",xlab="x_{n+1}",ylab="")
```

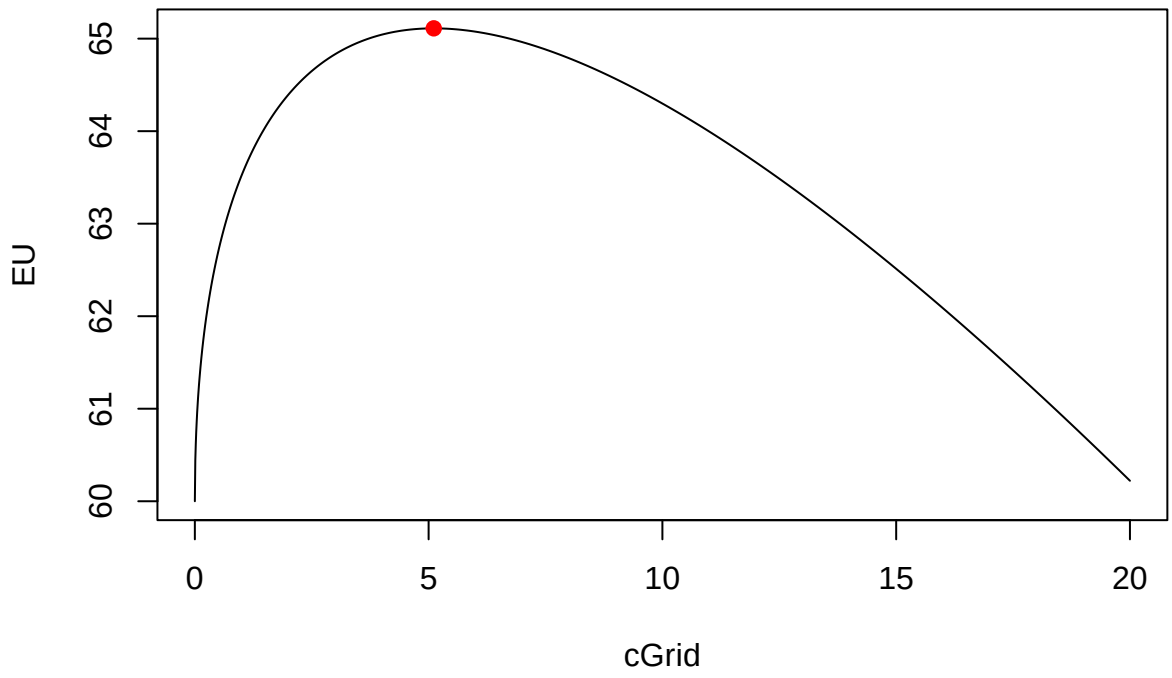


The posterior predictive distribution of x_{n+1} is plotted above.

3c

```
nIter <- 1e4
Mean_log_mu <- mean(log(rnorm(nIter,mean = 92,sd = 2)))
ExpectedUtility <- function(c, Mean_log_mu){
  EU <- 60 + sqrt(c)*Mean_log_mu - c
  return(EU)
}

cGrid <- seq(0,20,by = 0.01)
EU <- rep(NA,length(cGrid),1)
count <- 0
for (c in cGrid){
  count <- count + 1
  EU[count] = ExpectedUtility(c, Mean_log_mu)
}
plot(cGrid, EU, type = "l")
cOpt = cGrid[which.max(EU)] # This is the optimal c
points(cOpt,ExpectedUtility(c=cOpt, Mean_log_mu), col = "red",pch=19)
```




```
cOpt
```

```
## [1] 5.11
```

The optimal advertisement cost (c) is roughly 5.11 MSEK.