Laboration report in Bayesian Statistics

Bayesian Learning, Computer Lab 2

732A91

Duc Tran
William Wiik

Division of Statistics and Machine Learning
Department of Computer Science
Linköping University

26 April 2024

Question 1: Linear and polynomial regression

The dataset **Linkoping2022.xlsx** contains daily average temperatures (in degree Celcius) in Linköping over the course of the year 2022. The response variable is *temp* and the covariate *time* that you need to create yourself is defined by

$$time = \frac{\text{the number of days since the beginning of the year}}{365}$$

A Bayesian analysis of the following quadratic regression model is to be performed:

$$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2 + \epsilon, \epsilon \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$$

a)

Question: Use the conjugate prior for the linear regression model. The prior hyper parameters μ_0, Ω_0, ν_0 and σ_0^2 shall be set to sensible values. Start with $\mu_0 = (0, 100, -100)^T$, $\Omega_0 = 0.01 \cdot l_3$, $\nu_0 = 1$ and $\sigma_0^2 = 1$. Check if this prior agrees with your prior opinions by simulating draws from the joint prior of all parameters and for every draw compute the regression curve. This gives a collection of regression curves; one for each draw from the prior. Does the collection of curves look reasonable? If not, change the prior hyperparame ters until the collection of prior regression curves agrees with your prior beliefs about the regression curve.

Answer:

After some testing of different values for prior parameters we decided for the values as follows:

- $\mu_0 = (-5, 100, -100)$ - Our estimate of the prior
- $\Omega_0 = 1$ - The regularisation factor (1 is no regularisation)
- $\nu_0 = 2$ - "How much data we have in our prior knowledge", we are 2 in the group.
- $\sigma_0^2 = 5$ - How uncertain we are about the estimate of the prior.

```
# 1a                                                    ####
data <- read_xlsx("Linkoping2022.xlsx")
data$time <- (1:365)/365
n <- 365

# Prior
mu_0 <- c(-5, 100, -100)
omega_0 <- 1*diag(3)
nu_0 <- 2
sigma2_0 <- 5



# Simulating joint prior
set.seed(13)

# Step 1
X <- rchisq(10, n)
# Step 2
sample_sigma <- nu_0*sigma2_0/X
# Step 3
```

```r
beta_prior <- rmvnorm(10, mean = mu_0, sigma = sigma2_0 * solve(omega_0))


# Calculate values for the prior beta
covariates <-  cbind(rep(1,365), data$time, (data$time)^2)
y <- covariates %*% t(beta_prior)
plot_prior <- data.frame(time=data$time, y)
ggplot(plot_prior, aes(x=time)) +
  geom_line(aes(y=X1)) +
  geom_line(aes(y=X2)) +
  geom_line(aes(y=X3), colour="chartreuse4", size = 0.8) +
  geom_line(aes(y=X4)) +
  geom_line(aes(y=X5)) +
  geom_line(aes(y=X6), colour="chartreuse4", size = 0.8) +
  geom_line(aes(y=X7)) +
  geom_line(aes(y=X8)) +
  geom_line(aes(y=X9)) +
  geom_line(aes(y=X10)) +
  theme_bw() +
  ylim(-20, 30)
```

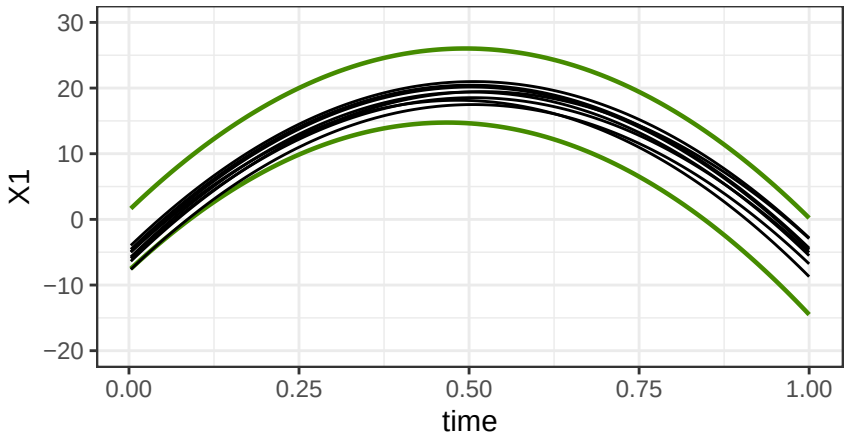

Figure 1: Simulated prior regression curves.

From figure 1, the two regression curves with lowest and highest values for temperature are highlighted in green. We believe that the regression curve should be somewhere in between these two regression curves and therefore the prior seems appropriate.

b)

Question: Write a function that simulate draws from the joint posterior distribution of β_0, β_1, β_2 and σ^2.

i. Plot a histogram for each marginal posterior of the parameters

ii. Make a scatter plot of the temperature data and overlay a curve for the posterior median of the regression function $f(time) = E[temp|time] = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2$, i.e. the median of $f(time)$ is computed for every value of *time*. In addition, overlay curves for the 90% equal tail posterior probability intervals of $f(time)$, i.e. the 5 and 95 posterior percentiles of $f(time)$ is computed for every value of time. Does the posterior probability intervals contain most of the data points? Should they?

Answer:

We are using a conjugate normal prior and we have the following:

Joint Prior for β and σ^2:

$$\beta|\sigma^2 \sim N\left(\mu_0, \sigma^2\Omega_0^{-1}\right)$$
$$\sigma^2 \sim Inv-\chi^2\left(\nu_0, \sigma_0^2\right)$$

Posterior

$$\beta|\sigma^2, y \sim N\left(\mu_n, \sigma^2\Omega_n^{-1}\right)$$
$$\sigma^2|y \sim Inv-\chi^2\left(\nu_n, \sigma_n^2\right)$$

where

$\mu_n = \left(X'X + \Omega_0\right)^{-1}\left(X'X\hat{\beta} + \Omega_0\mu_0\right)$

$\Omega_n = X'X + \Omega_0$

$\nu_n = \nu_0 + n$

$\nu_n\sigma_n^2 = \nu_0\sigma_0^2 + \left(y'y + \mu_0^{'}\Omega_0\mu_0 - \mu_n^{'}\Omega_n\mu_n\right)$

$\sigma_n^2 = \frac{\nu_0\sigma_0^2 + \left(y'y + \mu_0^{'}\Omega_0\mu_0 - \mu_n^{'}\Omega_n\mu_n\right)}{\nu_n}$

The code as follows simulates 10001 draws from the joint posterior distribution. The marginal posterior distribution for β_0, β_1, β_2, and σ^2 are presented in figure 2, 3, 4, and 5 respectively.

```
# 1b                                                ####
y <- data$temp
# OLS estimate of beta hat
beta_hat <- solve(t(covariates) %*% covariates) %*% t(covariates) %*% y

# Update of posterior parameters
mu_n <- solve(t(covariates) %*% covariates + omega_0) %*%
  (t(covariates) %*% covariates %*% beta_hat + omega_0 %*% mu_0)

omega_n <- t(covariates) %*% covariates + omega_0

nu_n <- nu_0 + n

nu_n_sigma2_n <- nu_0 * sigma2_0 +
  (t(y) %*% y + t(mu_0) %*% omega_0 %*% mu_0 - t(mu_n) %*% omega_n %*% mu_n)
```

```r
# Move nu_n to other side to get posterior sigma2_n
sigma2_n <- (nu_0 * sigma2_0 +
               (t(y) %*% y + t(mu_0) %*% omega_0 %*% mu_0 - t(mu_n) %*% omega_n %*% mu_n)) / nu_n


# Step 1
set.seed(13)
X <- rchisq(10001, n)
# Step 2
sample_sigma <- c(nu_n*sigma2_n)/X

beta_posterior <- matrix(nrow=10001, ncol=3)
for (i in 1:10001){
  beta_posterior[i,] <- rmvnorm(1, mean = mu_n, sigma = sample_sigma[i] * solve(omega_n))
}
sample_beta <- data.frame(beta_posterior)
colnames(sample_beta) <- c("beta_0", "beta_1", "beta_2")
```

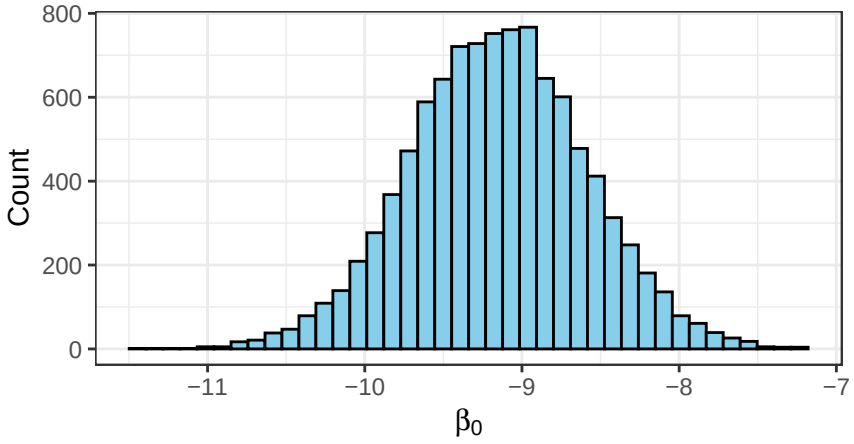

Figure 2: Posterior distribution for β_0

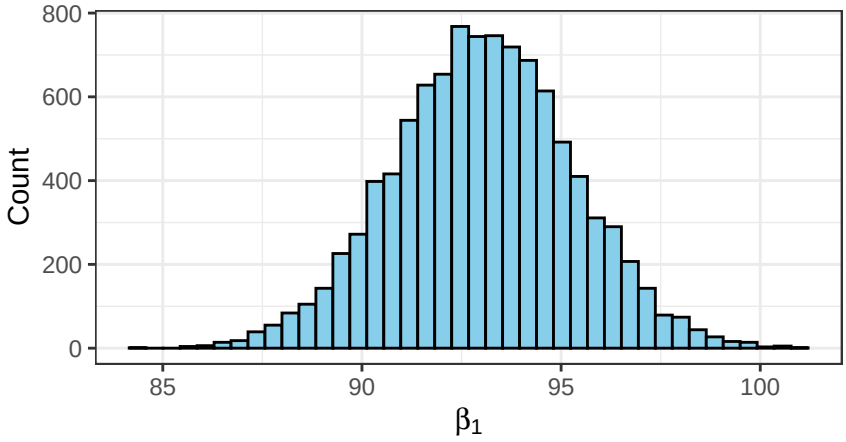

Figure 3: Posterior distribution for β_1

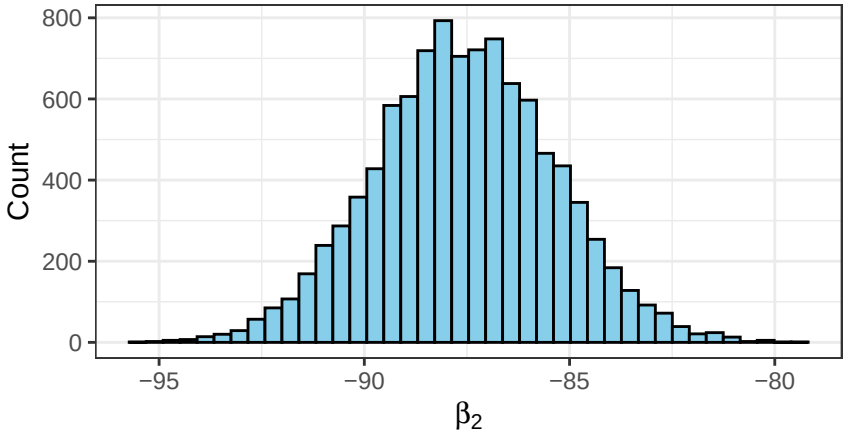

Figure 4: Posterior distribution for β_2

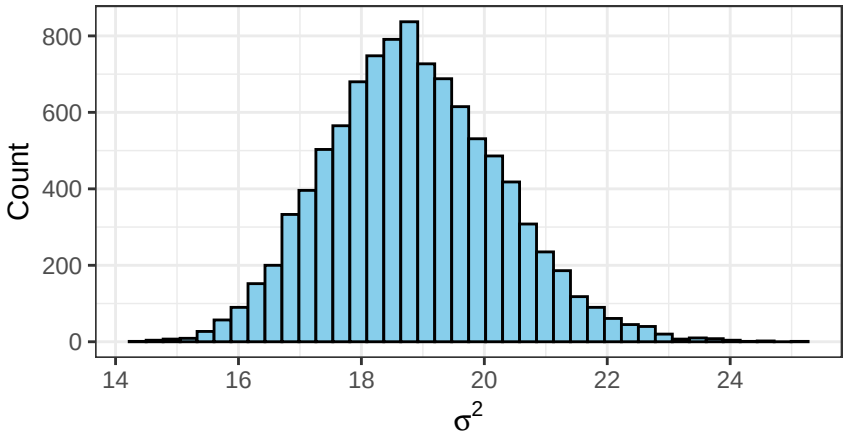

Figure 5: Posterior distribution for σ^2

In figure 6, the posterior median of the regression function for each time point with the observed temperature data are presented.

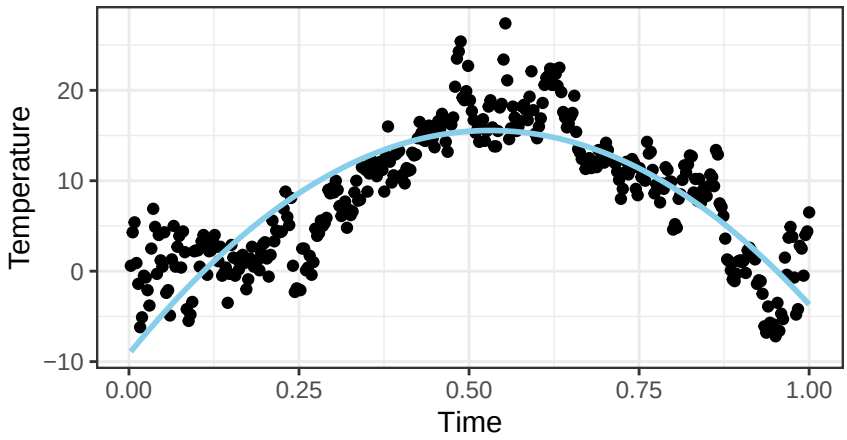

Figure 6: Scatter plot of the temperature data with the curve for the posterior median of the regression function.

In figure 7, the posterior median of the regression function for each time point with the observed temperature data and the 90% equal tail posterior intervals are presented.

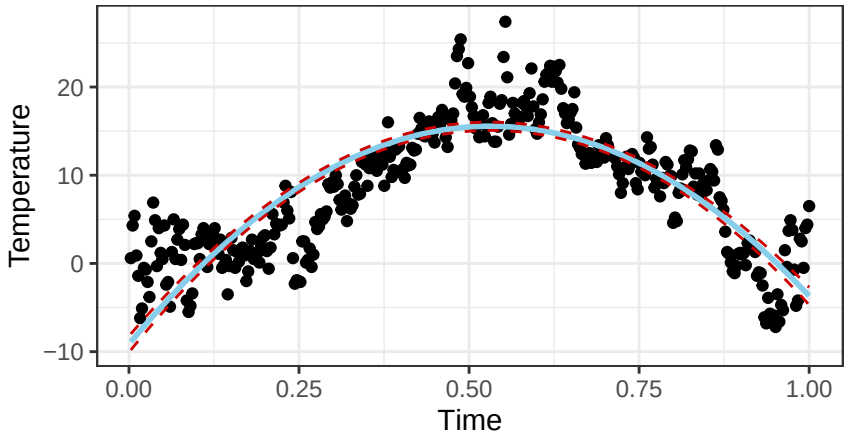

Figure 7: Scatter plot of the temperature data with the curve for the posterior median of the regression function(blue) and the 90% equal tail posterior interval(red).

From figure 7, the posterior probability interval does not contain most of the data points. With the 90% equal tail interval, the probability that the regression line lies in between this interval is 90%. So the interval does not need to contain most of the data points.

c)

Question: It is of interest to locate the time with the highest expected temperature (i.e. the time where $f(time)$ is maximal). Let's call this value $\tilde{x}$. Use the simulated draws in (b) to simulate from the posterior distribution of $\tilde{x}$.

Answer:

We have estimated the regression line:
$temp = \beta_0 + \beta_1 \cdot time + \beta_2 \cdot time^2$.

To find the maximum value for temp we can calculate the derivative with respect to time and solve for when the derivative is 0.

$$\frac{\partial temp}{\partial time} = \beta_1 + 2 \cdot \beta_2 \cdot time \tag{1}$$

Setting derivative to 0 and solving for time leads to equation as follows:

$$time = -\frac{\beta_1}{2 \cdot \beta_2} \tag{2}$$

The posterior distribution of $\tilde{x}$ is presented in figure 8.

```
# 1c                                                ####
time = data.frame(time=-sample_beta$beta_1 / (2*sample_beta$beta_2))

ggplot(time, aes(x=time)) +
  geom_histogram(colour="black", fill="skyblue", bins=70) +
  labs(x = "Time", y = "Count") +
  theme_bw()
```

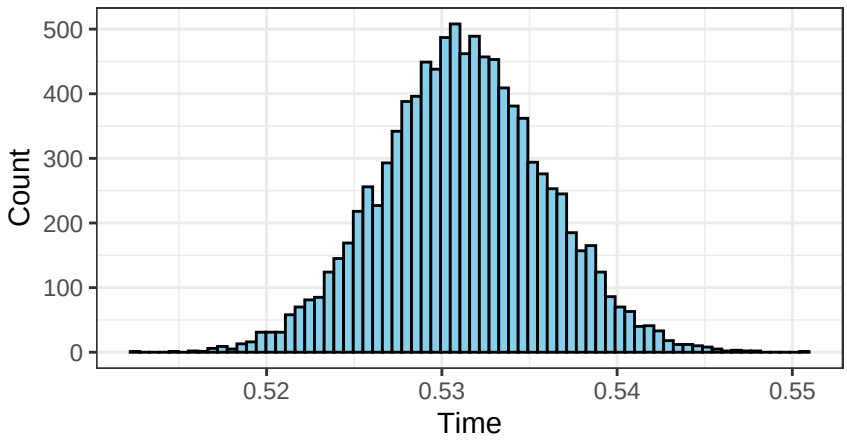

Figure 8: Posterior distribution of $\tilde{x}$

d)

Question:

Say now that you want to estimate a polynomial regression of order 10, but you suspect that higher order terms may not be needed, and you worry about overfitting the data. Suggest a suitable prior that mitigates this potential problem. You do not need to compute the posterior. Just write down your prior.

Answer:

For the higher terms that we suspect are not needed, we can set the prior mean (μ_0) of those coefficients to 0. Furthermore, we can have σ_0^2 close to 0 for those terms and have a large value for Ω_0, which is the regularisation factor. A combination of all these 3 parameters give an regularization effect on the higher terms. The prior could be implemented as follows.

```
# 1d                                                ####
# Prior
mu_0 <- c(-5, 100, -100, 0, 0, 0, 0, 0, 0, 0)
# Same prior values as before
omega_0 <- 1*diag(10)
# Change regularisation factor for higher terms
diag(omega_0)[4:10] <- 0.05
```

Question 2: Posterior approximation for classification with logistic regression

The dataset **WomenAtWork.dat** contains n = 132 observations on the following eight variables related to women: *Work*,*Constant*,*HusbandInc*, *EducYears*, *ExpYears*, *Age*, *NSmallChild* and *NBigChild*

a)

Question:

Consider the logistic regression model:

$$Pr(y = 1|\mathbf{x}, \beta) = \frac{exp(\mathbf{x}^T\beta)}{1 + exp(\mathbf{x}^T\beta)},$$

where y equals 1 if the woman works and 0 if she does not. $\mathbf{x}$ is a 7-dimensional vector containing the seven features (including a 1 to model the intercept). The goal is to approximate the posterior distribution of the parameter vector with a multivariate normal distribution

$$\beta|\mathbf{y}, \mathbf{x} \sim \mathcal{N}\left(\tilde{\beta}, J_y^{-1}(\tilde{\beta})\right)$$

where $\tilde{\beta}$ is the posterior mode and $J(\tilde{\beta}) = -\frac{\partial^2 \ln p(\beta|\mathbf{y})}{\partial\beta\partial\beta^T}$ is the negative of the observed Hessian evaluated at the posterior mode. Note that $\frac{\partial^2 \ln p(\beta|\mathbf{y})}{\partial\beta\partial\beta^T}$ is a 7x7 matrix with second derivatives on the diagonal and cross-derivative $\frac{\partial^2 \ln p(\beta|\mathbf{y})}{\partial\beta_i\partial\beta_j}$ on the off-diagonal. You can compute this derivative by hand, but we will let the computer do it numerically for you. Calculate both $\tilde{\beta}$ and $J\tilde{\beta}$ by using **optim** function in R. Use the prior $\beta \sim \mathcal{N}(0, \tau^2 I)$ where $\tau = 2$

Present the numerical values of $\tilde{\beta}$ and $J_y^{-1}(\tilde{\beta})$ for the **WomenAtWork** data. Compute an approximate 95% equal tail posterior probability interval for the regression coeffcient to the variable *NSmallChild*. Would you say that this feature is of importance for the probability that a woman works?

Answer:

The code used to estimate $\tilde{\beta}$ and $J_y^{-1}(\tilde{\beta})$ with optim and the result are as follows.

```
# 2a                                                    ####
data <- read.table("WomenAtWork.dat", header = TRUE)

Covs <- c(2:8) # Select which covariates/features to include
lambda <- 1 # scaling factor for the prior of beta
Nobs <- dim(data)[1] # number of observations
y <- data$Work

# Covariates
X <- as.matrix(data[,Covs])
Xnames <- colnames(X)
Npar <- dim(X)[2]

# Setting up the prior
mu <- as.matrix(rep(0,Npar)) # Prior mean vector
Sigma <- (4/lambda)*diag(Npar) # Prior covariance matrix
```

```
LogPostLogistic <- function(betas,y,X,mu,Sigma){
  linPred <- X%*%betas;
  logLik <- sum( linPred*y - log(1 + exp(linPred)) );
  logPrior <- dmvnorm(betas, mu, Sigma, log=TRUE);
  return(logLik + logPrior)
}


initVal <- matrix(0,Npar,1)
logPost = LogPostLogistic

OptimRes <- optim(initVal,logPost,gr=NULL,y,X,mu,Sigma,method=c("BFGS"),
                  control=list(fnscale=-1),hessian=TRUE)

names(OptimRes$par) <- Xnames # Naming the coefficient by covariates
approxPostStd <- sqrt(diag(solve(-OptimRes$hessian))) # Computing approximate standard deviations.
names(approxPostStd) <- Xnames # Naming the coefficient by covariates
print('The posterior mode is:')
```

```
## [1] "The posterior mode is:"
```

```
print(c(OptimRes$par))
```

```
##    Constant  HusbandInc   EducYears    ExpYears         Age NSmallChild
## -0.04036943 -0.03730689  0.17868950  0.12073637 -0.04618995 -1.47248930
##   NBigChild
## -0.02014458
```

```
print('The approximate posterior standard deviation is:')
```

```
## [1] "The approximate posterior standard deviation is:"
```

```
print(approxPostStd)
```

```
##    Constant  HusbandInc   EducYears    ExpYears         Age NSmallChild
##  1.38198486  0.02198474  0.08920960  0.03335982  0.02747315  0.47746764
##   NBigChild
##  0.16401959
```

We compared the result with the maximum likelihood estimates from the glmModel

```
# Result is similar to our result. Differ cause we have prior.
glmModel <- glm(Work ~ ., data = data[, -2], family=binomial)
glmModel$coefficients
```

```
## (Intercept)  HusbandInc   EducYears    ExpYears         Age NSmallChild
##  0.02262929 -0.03796308  0.18447411  0.12131763 -0.04858167 -1.56485140
##   NBigChild
## -0.02526059
```

Our result are similar to the glmModel.

The approximate 95% equal tail posterior probability is approximated with normal approximation for the regression coefficient NSmallChild.

```
# 95% equal tail.
# We do normal approximation.
# 95% equal tail for a normal distribution is +-2std
round(OptimRes$par[6] - 2 * approxPostStd[6], 4)
```

```
## NSmallChild
##     -2.4274
```

```
round(OptimRes$par[6] + 2 * approxPostStd[6], 4)
```

```
## NSmallChild
##     -0.5176
```

We have that approximate 95% equal tail posterior interval is between $[-2.4274, -0.5176]$. With 95% probability, this regression coefficient is in between these negative values. With an increase in NSmallChild with all other parameters being fixed, the probability of a woman working decreases.

b)

Question:

Use your normal approximation to the posterior from (a). Write a function that simulate draws from the posterior predictive distribution of $\Pr(y = 0|\mathbf{x})$, where the values of $\mathbf{x}$ corresponds to a 40-year-old woman, with two children (4 and 7 years old), 11 years of education, 7 years of experience, and a husband with an income of 18. Plot the posterior predictive distribution of $\Pr(y = 0|\mathbf{x})$ for this woman.

Answer:

We have that

$$Pr(y = 1|\mathbf{x}, \beta) = \frac{exp(\mathbf{x}^T\beta)}{1 + exp(\mathbf{x}^T\beta)}$$

which gives us that

$$Pr(y = 0|\mathbf{x}, \beta) = \frac{1}{1 + exp(\mathbf{x}^T\beta)}$$

With normal approximation, we use mode as the mean of the normal distribution and Jacobian as the variance with the general formula:

$$\theta|y \overset{approx}{\sim} N\Big(\tilde{\theta}, J_Y^{-1}(\tilde{\theta})\Big)$$

To simulate the posterior predictive distribution the following general algorithm is used:

1. Generate a **posterior draw** of $\theta\Big(\theta^{(1)}\Big)$ from $N\Big(\bar{y}, \sigma^2/n)\Big)$
2. Generate a **predictive draw** of $\tilde{y}\Big(\tilde{y}^{(1)}\Big)$ from $N\Big(\theta^{(1)}, \sigma^2\Big)$
3. Repeat step 1 and 2 N times for N number of predictive draws.

In our case we have:

- $\bar{y}$ = posterior modes from the optim function.
- $\sigma^2 = J_Y^{-1}(\tilde{\theta})$ from the optim function.

The posterior predictive distribution for this woman is presented in figure 9.

```
# 2b                                                    ####
# Normal approximation to the posterior
# Mean vector
mu <- OptimRes$par
# Sigma 2
sigma_2 <- solve(-OptimRes$hessian)
# Number of observations
n <- Nobs

# Sample 10000 different posterior draws
posterior_draws <- rmvnorm(10000, mean = mu, sigma = sigma_2/n)

# Posterior predictive draws
predictive_draws <- matrix(nrow=10000, ncol=7)
for (i in 1:dim(posterior_draws)[1]){
  predictive_draws[i,] <- rmvnorm(1, mean = posterior_draws[i,], sigma = sigma_2)
}



# Values for the woman.
x <- c(1, 18, 11, 7, 40, 1, 1)

# P(y=0|x, beta)
prob <- c(1/(1 + exp(x %*% t(predictive_draws))))
plot_data <- data.frame(prob)
ggplot(plot_data, aes(prob)) +
  geom_histogram(fill="darkorange3", colour="black", bins =50) +
  labs(x="Probability", y="Count") +
  theme_bw()
```

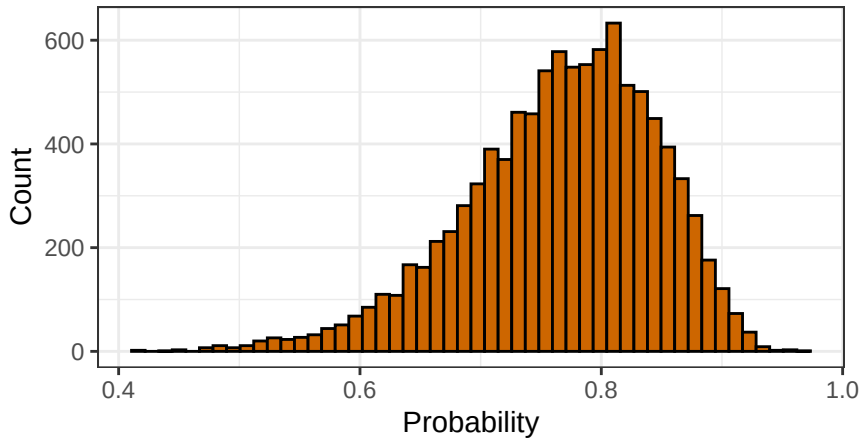

Figure 9: Posterior predictive distribution for the woman in question

c)

Question:

Now, consider 13 women which all have the same features as the woman in (b). Rewrite your function and plot the posterior predictive distribution for the number of women, out of these 13, that are not working.

Answer:

For each of the posterior predictive draw we get a probability θ which is the probability of a woman not working. We then simulate how many out of 13 women that are not working with the probability θ.

```
# 2c                                          ####
not_working <- c()
for(i in 1:length(prob)){
 not_working[i] <- rbinom(1, 13, prob[i])
}
```

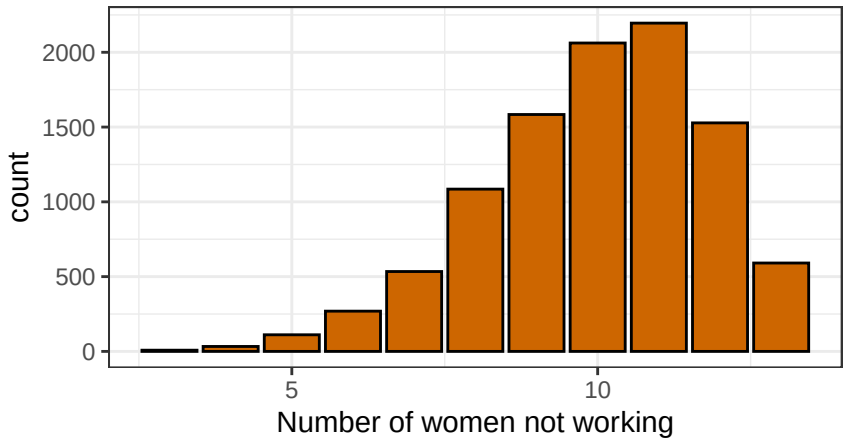

Figure 10: The posterior predictive distribution for the number of women, out of these 13, that are not working

Appendix

The code used in this laboration report are summarised in the code as follows:

```
library(ggplot2)
library(readxl)
library(mvtnorm)

knitr::opts_chunk$set(
  echo = TRUE,
  fig.width = 4.5,
  fig.height = 2.4)
# 1a                                          ####
data <- read_xlsx("Linkoping2022.xlsx")
data$time <- (1:365)/365
n <- 365

# Prior
mu_0 <- c(-5, 100, -100)
omega_0 <- 1*diag(3)
nu_0 <- 2
sigma2_0 <- 5



# Simulating joint prior
set.seed(13)

# Step 1
X <- rchisq(10, n)
# Step 2
sample_sigma <- nu_0*sigma2_0/X
# Step 3
beta_prior <- rmvnorm(10, mean = mu_0, sigma = sigma2_0 * solve(omega_0))


# Calculate values for the prior beta
covariates <-  cbind(rep(1,365), data$time, (data$time)^2)
y <- covariates %*% t(beta_prior)
plot_prior <- data.frame(time=data$time, y)
ggplot(plot_prior, aes(x=time)) +
  geom_line(aes(y=X1)) +
  geom_line(aes(y=X2)) +
  geom_line(aes(y=X3), colour="chartreuse4", size = 0.8) +
  geom_line(aes(y=X4)) +
  geom_line(aes(y=X5)) +
  geom_line(aes(y=X6), colour="chartreuse4", size = 0.8) +
  geom_line(aes(y=X7)) +
  geom_line(aes(y=X8)) +
  geom_line(aes(y=X9)) +
```

```
  geom_line(aes(y=X10)) +
  theme_bw() +
  ylim(-20, 30)


# 1b                                              ####
y <- data$temp
# OLS estimate of beta hat
beta_hat <- solve(t(covariates) %*% covariates) %*% t(covariates) %*% y

# Update of posterior parameters
mu_n <- solve(t(covariates) %*% covariates + omega_0) %*%
  (t(covariates) %*% covariates %*% beta_hat + omega_0 %*% mu_0)

omega_n <- t(covariates) %*% covariates + omega_0

nu_n <- nu_0 + n

nu_n_sigma2_n <- nu_0 * sigma2_0 +
  (t(y) %*% y + t(mu_0) %*% omega_0 %*% mu_0 - t(mu_n) %*% omega_n %*% mu_n)

# Move nu_n to other side to get posterior sigma2_n
sigma2_n <- (nu_0 * sigma2_0 +
               (t(y) %*% y + t(mu_0) %*% omega_0 %*% mu_0 - t(mu_n) %*% omega_n %*% mu_n)) / nu_n


# Step 1
set.seed(13)
X <- rchisq(10001, n)
# Step 2
sample_sigma <- c(nu_n*sigma2_n)/X

beta_posterior <- matrix(nrow=10001, ncol=3)
for (i in 1:10001){
  beta_posterior[i,] <- rmvnorm(1, mean = mu_n, sigma = sample_sigma[i] * solve(omega_n))
}
sample_beta <- data.frame(beta_posterior)
colnames(sample_beta) <- c("beta_0", "beta_1", "beta_2")



# 1b i                                            ####
ggplot(sample_beta, aes(x=beta_0)) +
  geom_histogram(fill="skyblue",colour="black", bins = 40) +
  labs(x = expression(beta[0]), y = "Count") +
  theme_bw()
ggplot(sample_beta, aes(x=beta_1)) +
  geom_histogram(fill="skyblue",colour="black", bins = 40) +
  labs(x = expression(beta[1]), y = "Count") +
```

```
  theme_bw()
ggplot(sample_beta, aes(x=beta_2)) +
  geom_histogram(fill="skyblue",colour="black", bins = 40) +
  labs(x = expression(beta[2]), y = "Count") +
  theme_bw()

plot_sigma <- data.frame(sigma = sample_sigma)
ggplot(plot_sigma, aes(x=sigma)) +
  geom_histogram(fill="skyblue",colour="black", bins = 40) +
  labs(x = expression(sigma^2), y = "Count") +
  theme_bw()

# 1b ii                                          ####

y <- matrix(ncol=365, nrow=10001)
for (i in 1:10001){
  y[i, ] <- sample_beta$beta_0[i ] + sample_beta$beta_1[i] * data$time + sample_beta$beta_2[i] * data$ti
}

plot_data <- data
plot_data$median <- apply(y, 2, FUN=median)

ggplot(plot_data, aes(x=time)) +
  geom_point(aes(y=temp), colour = "black") +
  geom_line(aes(y=median), colour = "skyblue", size=1) +
  labs(x = "Time", y = "Temperature") +
  theme_bw()
plot_data$quant_5 <- apply(y, 2, FUN=quantile, probs = 0.05)
plot_data$quant_95 <- apply(y, 2, FUN=quantile, probs = 0.95)

ggplot(plot_data, aes(x=time)) +
  geom_point(aes(y=temp), colour = "black") +
  geom_line(aes(y=median), colour = "skyblue", size=1) +
  geom_line(aes(y=quant_5), colour = "red3", linetype="dashed") +
  geom_line(aes(y=quant_95), colour = "red3", linetype="dashed") +
  labs(x = "Time", y = "Temperature") +
  theme_bw()


# 1c                                             ####
time = data.frame(time=-sample_beta$beta_1 / (2*sample_beta$beta_2))

ggplot(time, aes(x=time)) +
  geom_histogram(colour="black", fill="skyblue", bins=70) +
  labs(x = "Time", y = "Count") +
  theme_bw()


# 1d                                             ####
```

```r
# Prior
mu_0 <- c(-5, 100, -100, 0, 0, 0, 0, 0, 0, 0)
# Same prior values as before
omega_0 <- 1*diag(10)
# Change regularisation factor for higher terms
diag(omega_0)[4:10] <- 0.05
# 2a                                                ####
data <- read.table("WomenAtWork.dat", header = TRUE)

Covs <- c(2:8) # Select which covariates/features to include
lambda <- 1 # scaling factor for the prior of beta
Nobs <- dim(data)[1] # number of observations
y <- data$Work

# Covariates
X <- as.matrix(data[,Covs])
Xnames <- colnames(X)
Npar <- dim(X)[2]

# Setting up the prior
mu <- as.matrix(rep(0,Npar)) # Prior mean vector
Sigma <- (4/lambda)*diag(Npar) # Prior covariance matrix


LogPostLogistic <- function(betas,y,X,mu,Sigma){
  linPred <- X%*%betas;
  logLik <- sum( linPred*y - log(1 + exp(linPred)) );
  logPrior <- dmvnorm(betas, mu, Sigma, log=TRUE);
  return(logLik + logPrior)
}


initVal <- matrix(0,Npar,1)
logPost = LogPostLogistic

OptimRes <- optim(initVal,logPost,gr=NULL,y,X,mu,Sigma,method=c("BFGS"),
                  control=list(fnscale=-1),hessian=TRUE)

names(OptimRes$par) <- Xnames # Naming the coefficient by covariates
approxPostStd <- sqrt(diag(solve(-OptimRes$hessian))) # Computing approximate standard deviations.
names(approxPostStd) <- Xnames # Naming the coefficient by covariates
print('The posterior mode is:')
print(c(OptimRes$par))
print('The approximate posterior standard deviation is:')
print(approxPostStd)
# Result is similar to our result. Differ cause we have prior.
glmModel <- glm(Work ~ ., data = data[, -2], family=binomial)
glmModel$coefficients
# 95% equal tail.
```

```r
# We do normal approximation.
# 95% equal tail for a normal distribution is +-2std
round(OptimRes$par[6] - 2 * approxPostStd[6], 4)
round(OptimRes$par[6] + 2 * approxPostStd[6], 4)
# 2b                                          ####
# Normal approximation to the posterior
# Mean vector
mu <- OptimRes$par
# Sigma 2
sigma_2 <- solve(-OptimRes$hessian)
# Number of observations
n <- Nobs

# Sample 10000 different posterior draws
posterior_draws <- rmvnorm(10000, mean = mu, sigma = sigma_2/n)

# Posterior predictive draws
predictive_draws <- matrix(nrow=10000, ncol=7)
for (i in 1:dim(posterior_draws)[1]){
  predictive_draws[i,] <- rmvnorm(1, mean = posterior_draws[i,], sigma = sigma_2)
}



# Values for the woman.
x <- c(1, 18, 11, 7, 40, 1, 1)

# P(y=0|x, beta)
prob <- c(1/(1 + exp(x %*% t(predictive_draws))))
plot_data <- data.frame(prob)
ggplot(plot_data, aes(prob)) +
  geom_histogram(fill="darkorange3", colour="black", bins =50) +
  labs(x="Probability", y="Count") +
  theme_bw()
# 2c                                          ####
not_working <- c()
for(i in 1:length(prob)){
 not_working[i] <- rbinom(1, 13, prob[i])
}
plot_data <- data.frame(prob=c(not_working))
ggplot(plot_data, aes(prob)) +
  geom_bar(fill="darkorange3", colour="black") +
  labs(x="Number of women not working") +
 theme_bw()
```