Assignment2.R

snallfot

2023-04-26

# Assignment 2 Posterior approximation for classification with logistic regression  
#The dataset WomenAtWork.dat contains n = 132 observations on the following eight variables related to women:  
  
#2.a)   
#Consider the logistic regression model: Pr(y = 1|x, β) = exp(xTβ) / (1 + exp (xT β))  
# where y equals 1 if the woman works and 0 if she does not.  
#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 β with a multivariate normal distribution  
# β|y, x ∼ N(B\_hat, J^-1(B\_hat)) where B\_hat is the posterior mode and J(B\_hat) = −∂2 lnp(β|y)|  
# Note that ∂ β ∂ β T β = B\_hat is the negative of ∂2 ln p(β|y) the observed Hessian evaluated at the posterior mode.  
#Note that ∂β∂βT is a 7 × 7 matrix with second derivatives on the diagonal and cross-derivatives ∂2 ln p(β|y) ∂βi∂βj on the off-diagonal.   
#You can compute this derivative by hand, but we will let the computer do it numerically for you.   
#Calculate both B\_hat and J(B\_hat) by using the optim function in R.   
#[Hint: You may use code snippets from my demo of logistic regression in Lecture 6.] Use the prior β ∼ N(0,τ2I), where τ = 2.  
# Present the numerical values of β ̃ and J−1(β ̃) for the WomenAtWork dat  
#a. Compute an approximate 95% equal tail posterior probability interval for the regression coefficient to the variable NSmallChild.   
#Would you say that this feature is of importance for the probability that a woman works?  
#[Hint: You can verify that your estimation results are reasonable by comparing the posterior means  
# to the maximum likelihood estimates, given by: glmModel <- glm(Work ~ 0 + ., data = WomenAtWork, family = binomial).]  
  
  
#Import packages  
library(mvtnorm)  
  
#Read the data  
data <- (read.table("WomenAtWork.dat", header=TRUE))  
   
Xnames <- names(data[,2:ncol(data)]) #Names of columns  
head(data)

## Work Constant HusbandInc EducYears ExpYears Age NSmallChild NBigChild  
## 1 1 1 22.394940 12 7 43 0 3  
## 2 0 1 7.232000 8 10 34 0 7  
## 3 1 1 18.271990 12 4 41 1 5  
## 4 0 1 28.069000 14 2 43 0 2  
## 5 1 1 7.799889 12 10 31 0 1  
## 6 0 1 28.630000 16 6 37 0 3

n\_rows <- dim(data)[1] #number of observations  
  
  
LogPostLogistic <- function(betas,y,X,mu,Sigma){  
 linPred <- X%\*%betas;  
 logLik <- sum( linPred\*y - log(1 + exp(linPred)) );  
 #if (abs(logLik) == Inf) logLik = -20000; # Likelihood is not finite, stear the optimizer away from here!  
 logPrior <- dmvnorm(betas, mu, Sigma, log=TRUE);  
   
 return(logLik + logPrior)  
}  
  
#initialize variables  
set.seed(123)  
y <- as.numeric(data[,1]) #target output  
X <- as.matrix(data[,-1]) #training data  
X <- apply(X, c(1, 2), as.numeric) #convert from characters to doubles  
n\_col <- dim(X)[2] #number of columns in training data  
tao = 2 #prior std.dev for Beta  
initVal <- matrix(0,n\_col,1) #initial values for Beta  
mu <- as.matrix(rep(0,n\_col)) # Prior mean vector for Beta  
Sigma <- tao^2\*diag(n\_col) # Prior covariance matrix for Beta  
logPost <- LogPostLogistic; #function to be minimized  
  
OptimRes <- optim(initVal,logPost,gr=NULL,y,X,mu,Sigma,method=c("BFGS"),control=list(fnscale=-1),hessian=TRUE)  
  
# Printing the results of mean and variance of normally distributed Beta   
OptimBeta <- OptimRes$par  
names(OptimBeta) <- Xnames # Naming the coefficient by covariates  
PostCov <- solve(-OptimRes$hessian) #get covariance matrix by inverting the negative hessian  
approxPostStd <- sqrt(diag(PostCov)) # Computing approximate standard deviations.  
names(approxPostStd) <- Xnames # Naming the coefficient by covariates  
print('The posterior mode is:')

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

print(OptimBeta)

## [,1]  
## [1,] -0.04036943  
## [2,] -0.03730689  
## [3,] 0.17868950  
## [4,] 0.12073637  
## [5,] -0.04618995  
## [6,] -1.47248930  
## [7,] -0.02014458  
## attr(,"names")  
## [1] "Constant" "HusbandInc" "EducYears" "ExpYears" "Age"   
## [6] "NSmallChild" "NBigChild"

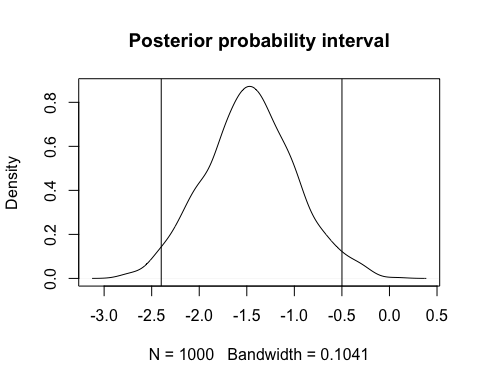
print('The approximate posterior standard deviation is:')

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

print(approxPostStd)

## Constant HusbandInc EducYears ExpYears Age NSmallChild   
## 1.38198487 0.02198474 0.08920960 0.03335982 0.02747315 0.47746764   
## NBigChild   
## 0.16401959

#Draw samples from distribution of NSmallChild  
my <- OptimBeta[6] #mean of Beta[NSmallChild]  
sigma <-approxPostStd[6] #std.dev of Beta[NSmallChild]  
samples <- rnorm(1000, my, sigma)  
quantiles <- quantile(samples, c(0.025, 0.975))  
plot(density(samples), main="Posterior probability interval")  
abline(v=quantiles)



print("Upper and lower bounds are:")

## [1] "Upper and lower bounds are:"

print(quantiles)

## 2.5% 97.5%   
## -2.3995187 -0.4994643

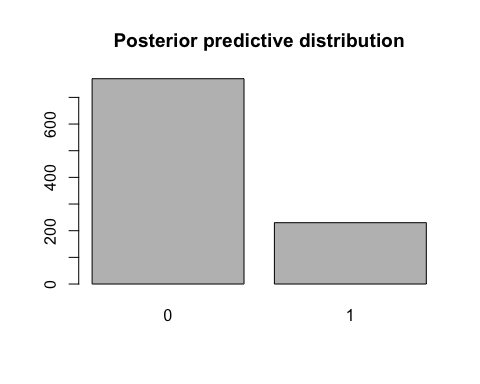
#Check that estimation results are reasonable  
glmModel <- glm(Work ~ 0 + ., data = data, family = binomial)  
print(OptimBeta)

## [,1]  
## [1,] -0.04036943  
## [2,] -0.03730689  
## [3,] 0.17868950  
## [4,] 0.12073637  
## [5,] -0.04618995  
## [6,] -1.47248930  
## [7,] -0.02014458  
## attr(,"names")  
## [1] "Constant" "HusbandInc" "EducYears" "ExpYears" "Age"   
## [6] "NSmallChild" "NBigChild"

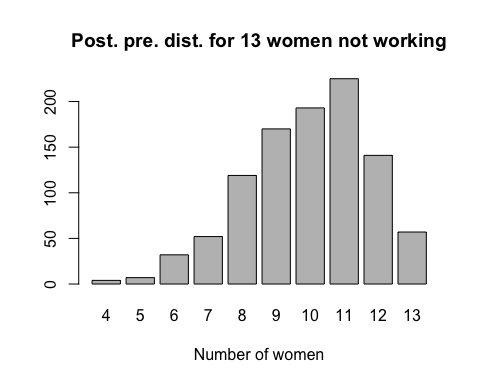
print(glmModel$coefficients)

## Constant HusbandInc EducYears ExpYears Age NSmallChild   
## 0.02262929 -0.03796308 0.18447411 0.12131763 -0.04858167 -1.56485140   
## NBigChild   
## -0.02526059

#values are similar, estimation seems reasonable. Since the boundary is negative, it seems like a higher number   
#of small children makes a woman less likely to work.   
  
#b) Use your normal approximation to the posterior from (a).   
#Write a function that simulate draws from the posterior predictive distribution of Pr(y = 0|x),   
#where the values of 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|x) for this woman.  
#[Hints: The R package mvtnorm will be useful. Remember that Pr(y = 0|x) can be calculated for each posterior draw of β.]  
  
#sigmoid function  
sigmoid <- function(x, beta) {  
 s <- beta %\*% x  
 return (exp(s) / (1+exp(s)))  
}  
  
set.seed(123)  
PostStd <- c(approxPostStd) #convert to vector  
beta\_samples <- rmvnorm(n=1000, mean=OptimBeta, sigma=PostCov) #draw samples of Beta from the distribution  
x <- c(1, 18, 11, 7, 40, 1, 1) #values for a single sample of x  
  
y <- rep(0,1000)  
for(i in 1:1000) {  
 y[i] = sigmoid(x, beta\_samples[i,])   
}  
  
#draw random binomial samples given the probabilities  
predictions <- rep(0,1000)  
for(i in seq\_along(y)) {  
 p = y[i] #probability  
 predictions[i] <- rbinom(1,1,p)   
}  
barplot(table(predictions), main="Posterior predictive distribution")



#We can see that the prediction is 0 for approximately 80% of the samples, which means it is  
#likely that this woman does not work.  
  
#c) 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.   
#[Hint: Simulate from the binomial distribution, which is the distribution for a sum of Bernoulli random variables.]  
set.seed(123)  
n\_women <- 13  
y <- rep(0,1000)  
predictions <- c()  
  
#samples from Beta distribution 1000 times and make predictions. Simulate from binomial distribution for the 13 women.   
for (i in 1:1000) {  
 beta\_samples <- rmvnorm(n=1000, mean=OptimBeta, sigma=PostCov) #draw samples of Beta from the distribution  
 y[i] <- sigmoid(x, beta\_samples[i,]) #probabilities  
 predictions <- c(predictions, n\_women-rbinom(n=1, size=n\_women, y[i])) #simulate from binomial distribution where each result is prediction of number of women working  
}  
  
barplot(table(predictions), main=paste("Post. pre. dist. for", n\_women, "women not working"), xlab="Number of women")



#The conclusion is that according to the posterior predictive distribution, it seems that 11 women most likely does not work, with high numbers  
#for 8-12 women not working.