732A91: Lab 2
Bayesian Learning

Sarah Alsaadi, Carles Sans Fuentes

May 22, 2017

Linear and polynomial regression

In this exercise we use the dataset TempLinkoping.txt in our analysis. The dataset contains daily temperatures (in Celsius degrees) at Malmslatt and Linkoping over the course of the year 2016, having as a response variable temp and covariate time.

First we determine the prior distribution of the model parameter. Given that our likelihood is a quadratic regression (see the summary of the quadratic regression model below) and that its conjugate prior is a multivariate normal distribution (because there is more than one Beta parameter), the prior hyperparameters are chosen as following:

1. μ_0 is the linear coefficients from our data (our best guess is just the computation of the parameters as if it was a quadratic regression of our data),
2. ω_0 a diagonal matrix of 1s given that all data has the same importance,
3. v_0 equal 6 in order to not to give too much importance to our prior and
4. σ_0^2 equal 16 (similar variance of the quadratic regression model)

```
> summary(lm)

Call:
lm(formula = temp ~ time + I(time^2), data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-10.0408  -2.6971  -0.1414   2.5157  12.2085

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.6754     0.6475  -16.49   <2e-16 ***
time         93.5980     2.9822   31.39   <2e-16 ***
I(time^2)   -85.8311     2.8801  -29.80   <2e-16 ***
---

Residual standard error: 4.107 on 363 degrees of freedom
Multiple R-squared:  0.7318,  Adjusted R-squared:  0.7304
F-statistic: 495.3 on 2 and 363 DF,  p-value: < 2.2e-16
```

In order to check whether our prior is sensible, we have simulated 1000 draws from the joint prior of all parameters and for every draw compute the regression curve. In order to check whether our prior look properly, we have evaluated the distributions of the Beta parameters and comparing it with the one from the quadratic regression. Below we have the mean of the beta coefficients.

```
apply(betasprior, 2, mean)
     beta0      beta1      beta2
 -9.991929  92.984489 -85.085894
```

It can be seen that given that our prior is really similar to our data, the Beta mean coefficients are almost the same.

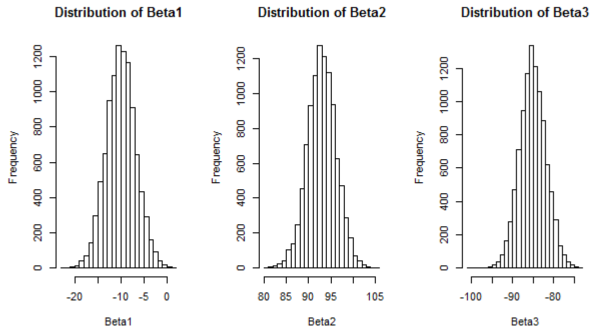

Figure 1: Histogram plot distribution of the our 3 Beta prior after 10000 simulations

It can be seen that the distribution looks normally distributed through the mean parameter of our data. This is good given the prior used. For this reason, we can say the curve looks reasonable.

Now, we have written a program that simulates from the joint posterior distribution of $\beta_0, \beta_1, \beta_2 and \sigma_2$. We have produced a scatter plot of the temperature data and overlay a curve for the posterior mean of the regression function $f(time) = \beta_0 + \beta_1 * time + \beta_2 * time^2$ as well as the 95% equal tail posterior probability intervals for every value of time and then connect the lower and upper limits of the interval by curves (see figure 2)

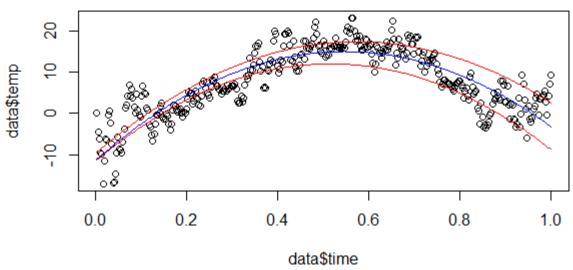

Figure 2: Histogram of the data with the posterior mean distribution of the data (Blue line) and its 95% credible interval (red lines)

It can be seen that the variance of the data increases for the last half of the prediction.

In 1d we are asked to locate the time with the highest expected temperature (that is, the time where E(temp—time) is maximal).We have used the previous betas to simulate the new data from the highest value of temp given time and we have plot the distribution of this points (see figure 3 below). It is seen that the mean of the maximum point is around 14-16. This makes sense since it is similar to the data we got.

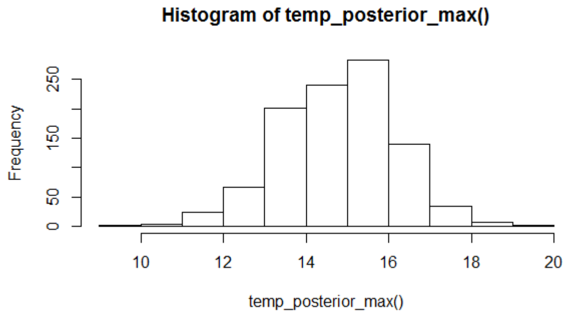

Figure 3: Scatterplot of the data together with the computed tempreature (blue curve) from the posterior mean of $\beta_0, \beta_1, \beta_2$ and the lower 2.5% and the upper 97.5% posterior credible interval

Finally, we are asked to estimate a polynomial model of order 7 and we are told that higher order terms may not be needed. Since we have a strong belief that higher order terms are not needed, we specify the prior parameters for those coefficients to reflect that. This would be a an expected value close to 0 and a very small variance.

Question 2

In activity 2 we are asked to write the model with the logistic function a data set to predict the probability that a woman will work given some variables that defines her.

For that, first we fit our data to the logistic model directly, and we get the following:

```
> summary(glmModel)

Call:
glm(formula = Work ~ 0 + ., family = binomial, data = Womenwork)

Deviance Residuals:
    Min       1Q    Median       3Q      Max
-2.1662  -0.9299   0.4391   0.9494   2.0582

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
Constant     0.64430    1.52307   0.423 0.672274
HusbandInc  -0.01977    0.01590  -1.243 0.213752
EducYears    0.17988    0.07914   2.273 0.023024 *
ExpYears     0.16751    0.06600   2.538 0.011144 *
ExpYears2   -0.14436    0.23585  -0.612 0.540489
Age         -0.08234    0.02699  -3.050 0.002285 **
NSmallChild -1.36250    0.38996  -3.494 0.000476 ***
NBigChild   -0.02543    0.14172  -0.179 0.857592
---

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.26  on 200  degrees of freedom
Residual deviance: 222.73  on 192  degrees of freedom
AIC: 238.73

Number of Fisher Scoring iterations: 4
```

Now we are asked to approximate the posterior distribution of the 8-dim parameter vector β with a multivariate normal distribution. For that, we have implemented a logistic function and used the optim() from R to find the Hessian and the best parameters for the best case given optimizing the posterior. The results is the following one:

```
> OptimResults
$par
[1] -0.020734822  0.198537870  0.170970267 -0.157370056 -0.073756571 -1.308417158
[7]  0.002373846

$value
[1] -134.015
```

```

$counts
function gradient
      59       14

$convergence
[1] 0

$message
NULL

$hessian
            [,1]        [,2]         [,3]         [,4]        [,5]        [,6]
[1,] -21691.9214 -10335.9656  -7724.06469 -1095.904557 -33606.6413 -190.535283
[2,] -10335.9656  -6069.9347  -4546.30708  -647.404832 -19675.4015 -127.202981
[3,]  -7724.0647  -4546.3071  -5353.80184  -982.743357 -16236.0124  -75.200938
[4,]  -1095.9046   -647.4048   -982.74336  -214.881014  -2470.3813   -7.716988
[5,] -33606.6413 -19675.4015 -16236.01237 -2470.381294 -68796.8982 -321.680446
[6,]   -190.5353   -127.2030    -75.20094    -7.716988   -321.6804  -11.942247
[7,]   -996.0942   -584.7677   -373.77395   -42.860366  -1894.1299  -12.925833
            [,7]
[1,]  -996.09418
[2,]  -584.76769
[3,]  -373.77395
[4,]   -42.86037
[5,] -1894.12992
[6,]   -12.92583
[7,]  -123.33156
```

Also, the 95% CI for the variable NSmallChild must be found (see below) in order to define whether the variable is important or not

```
lowbound highbound
-2.028515 -0.588319
```

It can be seen that the credible interval is far from 0, which means that the variable is an important feature.

Finally, we are asked to Write a function that simulates from the predictive distribution of the response variable in a logistic regression.We have used our previously normal approximation to simulate and plot the predictive distribution for the Work variable for a 40 year old women, with two children (3 and 9 years old), 8 years of education, 10 years of experience. and a husband with an income of 10. The results is the following histogram:

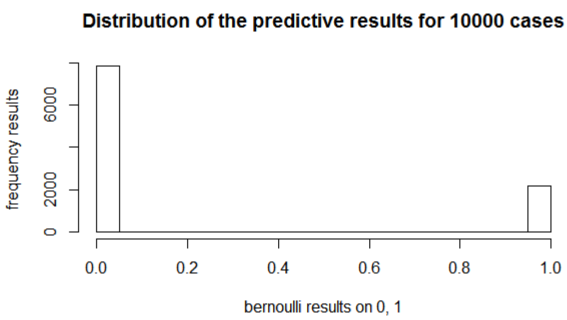

Figure 4: Histogram of the predictive distribution of the response variable in a logistic regression

The result shows that it is quite much probable that the Woman will not be able to work.

Contributions

All results and comments presented have been developed and discussed together by the members of the group.

Appendix

Question 1

```
#lab 2 Bayesian learning

# install.packages("mvtnorm")
# install.packages("MASS")
library("mvtnorm")
library("geoR")
library("MASS")

data<-read.table("~/Google Drive/Kurser/Bayesian learning/Lab2/TempLinkoping2016.txt",head=TRUE
    )
data<- read.csv("C:/Users/Carles/Desktop/Bayesian learning/Part2/TempLinkoping2016.txt", sep ="
    \t")

#1.a
lm<-lm(temp~time+I(time^2),data)
summary(lm)

#1.b
mu_0<-c(-10,93,-85)
sigma_0<-16
v_0<-6
omega_0<-diag(3)
sigma_prior<-rinvchisq(1,df=v_0,scale=sigma_0)
beta_prior<-rmvnorm(n=1, mean =mu_0, sigma = sigma_prior*ginv(omega_0))
regress_prior<-beta_prior[1]+beta_prior[2]*data$time+beta_prior[3]*(data$time)^2
plot(y=data$temp,x=data$time)
lines(y=regress_prior,x=data$time,col="blue")

betasprior<-function(n)
{
  beta0<-numeric(n)
  beta1<-numeric(n)
  beta2<-numeric(n)

  for (i in 1:n)
  {
    beta0[i]<-as.vector(rmvnorm(n=1, mean =mu_0, sigma = sigma_prior*ginv(omega_0)))[1]
    beta1[i]<-as.vector(rmvnorm(n=1, mean =mu_0, sigma = sigma_prior*ginv(omega_0)))[2]
    beta2[i]<-as.vector(rmvnorm(n=1, mean =mu_0, sigma = sigma_prior*ginv(omega_0)))[3]

     }
  return(data.frame(beta0, beta1, beta2))
}
betasprior<-betasprior(10000)
apply(betasprior, 2, mean)
dim(betasprior)
par(mfrow=c(1,3))

hist(betasprior[,1], breaks = 30, xlab = "Beta1", main = "Distribution of Beta1")
hist(betasprior[,2], breaks = 30, xlab = "Beta2", main = "Distribution of Beta2")
hist(betasprior[,3], breaks = 30, xlab = "Beta3", main = "Distribution of Beta3")



#1.c
X<-cbind(rep(1,nrow(data)),data$time,(data$time)^2)
#beta_hat<-ginv(t(X)%*%X)%*%t(X)%*%data$temp
mu_n<-ginv(t(X)%*%X+omega_0)%*%(t(X)%*%data$temp+omega_0%*%mu_0)
Sigma_n<-t(X)%*%X+omega_0
v_n<-v_0+nrow(data)
sigma_n<-(1/v_n)*(v_0*sigma_0+(t(data$temp)%*%data$temp+t(mu_0)%*%omega_0%*%mu_0-t(mu_n)%*%
    Sigma_n%*%mu_n))

sigma_posterior<-as.vector(rinvchisq(1,df=v_n,scale=sigma_n))
beta_posterior<-rmvnorm(n=1, mean =mu_n, sigma = sigma_posterior*ginv(Sigma_n))
regress_posterior<-beta_posterior[1]+beta_posterior[2]*data$time+beta_posterior[3]*(data$time)
    ^2
plot(y=data$temp,x=data$time)
lines(y=regress_posterior,x=data$time,col="blue")

betas<-function(n)
{
  beta0<-numeric(n)
  beta1<-numeric(n)
  beta2<-numeric(n)

  for (i in 1:n)
  {
    beta0[i]<-as.vector(rmvnorm(n=1, mean =mu_n, sigma = sigma_posterior*ginv(Sigma_n)))[1]
    beta1[i]<-as.vector(rmvnorm(n=1, mean =mu_n, sigma = sigma_posterior*ginv(Sigma_n)))[2]
```

```
    beta2[i]<-as.vector(rmvnorm(n=1, mean =mu_n, sigma = sigma_posterior*ginv(Sigma_n)))[3]

    }
  return(data.frame(beta0,beta1, beta2))
  }


betas<-betas(1000)
betas0<-betas[,1]
betas1<-betas[,2]
betas2<-betas[,3]


temp_posterior<-function()
{
  n=1000
  k=366
  temp_posterior= matrix(data=NA, nrow=n, ncol=k)
  for(j in 1:k){
    for(i in 1:n){
      temp_posterior[i,j] = betas0[i]+betas1[i]*data$time[j]+betas2[i]*(data$time[j])^2
    }
  }
  return(temp_posterior)
  }
temp_posterior<-temp_posterior()

quantile_f<-function()
{
  k=366
  q_lower<-numeric(k)
  q_upper<-numeric(k)

  for (i in 1:k)
  {
    q_lower[i]=quantile(temp_posterior[,i],  probs = c(0.025, 0.975))[1]
    q_upper[i]=quantile(temp_posterior[,i],  probs = c(0.025, 0.975))[2]

    }
  return(data.frame(q_lower,q_upper))
}

quantile_f<-quantile_f()
plot(q_lower,type="l")


par(mfrow= c(1,1))
plot(y=data$temp,x=data$time)
lines( y =quantile_f[,2],x=data$time, col= "red")
lines(y =quantile_f[,1],x=data$time, col = "red")
lines(y=regress_posterior,x=data$time,col="blue")


myfunc<- function (data){
  regress_posterior<-betas[1]+betas[2]*data+betas[3]*(data)^2
  return(regress_posterior)

}

temp_mean<-mean(beta_posterior[,1])+mean(beta_posterior[,2])*data$time+mean(beta_posterior[,3])
     *data$time^2
t<-c(timeMax=data$time[which.max(temp_mean)], tempMax=max(temp_mean))


temp_posterior_max<-function()
{
  n=1000

  temp_posterior_max= numeric(n)

    for(i in 1:n){
      temp_posterior_max[i] = betas[i,1]+betas[i,2]*t[1]+betas[i,3]*t[1]^2
    }

  return(temp_posterior_max)
}

hist(temp_posterior_max())

##1d

lm7<-lm(temp~time+I(time^2)+I(time^3)+I(time^4)+I(time^5)+I(time^6)+I(time^7),data)
mu7<-lm7$coefficients
sigma7<-16
v7<-365
#First values
lambdaplot<-function(lambda){
```

```
  lambda <-0.5
  Sigma7<-diag(8)*lambda
  sigma_prior7<-rinvchisq(1,df=v7,scale=sigma7)
  regress_prior7<- matrix(nrow = 1000, ncol = 366)
  for(i in 1:1000){
  beta_prior7<-rmvnorm(n=1, mean =mu7, sigma = sigma_prior7*ginv(Sigma7))
  regress_prior7[i,]<-beta_prior7[1]+beta_prior7[2]*data$time+beta_prior7[3]*(data$time)^2+beta_prior7[4]*(data$time)^3+beta_prior7[5]*(data$time)^4+beta_prior7[6]*(data$time)^5+beta_prior7[7]*(data$time)^6+beta_prior7[8]*(data$time)^7

  }
  return(colMeans(regress_prior7))
}
lambda05<-lambdaplot(lambda =0.5)
lambda1<-lambdaplot(lambda =1)
lambda5<-lambdaplot(lambda =10)
lambda10<-lambdaplot(lambda =100)

#lambda<-seq(from=0,to=10,by=1)
par(mfrow = c(2,2))
plot(y=data$temp,x=data$time, main = "lambda = 0.5")
lines(y=lambda05,x=data$time,col="red")
plot(y=data$temp,x=data$time, main = "lambda = 1")
lines(y=lambda1,x=data$time,col="red")
plot(y=data$temp,x=data$time, main = "lambda = 10")
lines(y=lambda5,x=data$time,col="red")
plot(y=data$temp,x=data$time, main = "lambda = 100 ")
lines(y=lambda10,x=data$time,col="red")
```

Question 2

```

#lab 2 Bayesian learning

#######################################################################
####2a
Womenwork<-read.table("~/Google Drive/Kurser/Bayesian learning/Lab2/WomenWork.dat.txt",head=TRUE)
Womenwork<- read.table("C:/Users/Carles/Desktop/Bayesian learning/Part2/WomenWork.dat.txt",head=TRUE)


glmModel <-   glm(Work ~ 0 + ., data = Womenwork, family = binomial)

summary(glmModel)

######b
#install.packages("mvtnorm") # Loading a package that contains the multivariate normal pdf
library("mvtnorm") # This command reads the mvtnorm package into R's memory. NOW we can use dmvnorm function.

# Loading data from file
#Data<-read.csv("/home/carsa564/Desktop/Bayesian learning/WomenWork.dat.txt", sep = "")
Data<- read.csv("C:/Users/M/Desktop/Statistics and Data Mining Master/Semester 2/Bayesian Learning/lab2/WomenWork.dat.txt", sep = "")
Data<-read.csv("~/Google Drive/Kurser/Bayesian learning/Lab2/WomenWork.dat.txt",sep ="")
Data<- read.csv("C:/Users/Carles/Desktop/Bayesian learning/Part2/WomenWork.dat.txt",sep ="")

tau <- 10;         # Prior scaling factor such that Prior Covariance = (tau^2)*I
chooseCov <- c(2:8)  # Here we choose which covariates to include in the model

y <- as.vector(Data[,1]); # Data from the read.table function is a data frame. Let's convert y and X to vector and matrix.
X <- as.matrix(Data[,2:ncol(Data)]);
covNames <- names(Data)[2:length(names(Data))];
X <- X[,chooseCov]; # Here we pick out the chosen covariates.
covNames <- covNames[chooseCov];
nPara <- dim(X)[2];

# Setting up the prior
mu <- as.vector(rep(0,nPara)) # Prior mean vector
Sigma <- tau^2*diag(nPara);




LogPostLogistic <- function(betaVect,y,X,mu,Sigma){

  nPara <- length(betaVect);
  linPred <- X%*%betaVect;

  # evaluating the log-likelihood
  logLik <- sum( linPred*y -log(1 + exp(linPred)));
```

```
  if (abs(logLik) == Inf) logLik = -20000; # Likelihood is not finite, stear the optimizer away
       from here!

  # evaluating the prior
  logPrior <- dmvnorm(betaVect, matrix(0,nPara,1), Sigma, log=TRUE);

  # add the log prior and log-likelihood together to get log posterior
  return(logLik + logPrior)
}

initVal <- as.vector(rep(0,dim(X)[2]));

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

BetaCoef<- OptimResults$par
J<--solve(OptimResults$hessian)

myconf95<-c(lowbound =BetaCoef[6]-1.96*sqrt(J[6,6]), highbound=BetaCoef[6]+1.96*sqrt(J[6,6]))



######2c

##J as the matrix of covariates for sigma is calculated before as well as BetaCoef which is mu
     or the BetaCoefficients
Woman<-c(10, 8, 10, 100, 40, 2,0)




logProv<- function(n, betas= BetaCoef, covariances= J,obs=Woman){
  prob<- integer(n)
  Pr<- integer(n)
  for(i in 1:n){
    myrandombetas<- as.vector(rmvnorm(1, mean= BetaCoef, sigma = covariances))
    Pr[i]<- exp((obs)%*%myrandombetas)/(1 + exp((obs)%*%myrandombetas))
    prob[i]<- rbinom(1,1,Pr[i])
  }

  return(list("Predictive results"= prob, "Probability results"= Pr))
}


distribution_of_the_predictive_results<- logProv(10000,  betas= BetaCoef, covariances= J,obs=
     Woman)

hist(distribution_of_the_predictive_results[[1]],
     xlab = "bernoulli results on 0, 1",
     ylab = "frequency results",
     main = "Distribution of the predictive results for 10000 cases")
```