lab2_code

Akshath Srinivas (akssr921) Yaning Wang (yanwa579)

```r
#question 1 a

#get data and create covariate(time) and y(temp)
data<-readxl::read_xlsx('Linkoping2022.xlsx')
temp<-as.data.frame(data[,3])
temp<-temp$temp
time<-as.numeric(as.Date(data$datetime) - as.Date("2022-01-01"))/365

#creating X matrix
#x value for intercept
X_mat <- matrix(rep(1,length(time)))
#cbing time and time^2
X_mat<-cbind(X_mat,time,time^2)

#sigma(tau) and mu for sigma_2 simulation
y<-time
mu<-1
tau<-1

#mu_0 and omega_0 to get beta values
mu_0<-c(0, 100, -100)
omega_0<-solve(0.01 * diag(3))

#draw joint prior using inversechisquare to draw sigma_2 and using
#sigma_2,mu_0 and omega_0 draw for beta values
#creating a function to change the values of the parameters
joint_prior_func<-function(n,x,y,mu,tau,mu_0,omega_0){
  joint_prior<-matrix(NA, ncol = 3, nrow = n)
  for(i in 1:n){
    sigma_2<-LaplacesDemon::rinvchisq(1,mu,tau)
    joint_prior_vec <- mvtnorm::rmvnorm(n = 1, mean = mu_0, sigma = omega_0*sigma_2)
    joint_prior[i,]<-joint_prior_vec
  }
  #get predictions
  pred<-X_mat%*%t(joint_prior)
  return(pred)
}
#get prediction
prediction<-joint_prior_func(50,X_mat,y,mu,tau,mu_0,omega_0)
#get_first 20 predictions to plot
prediction_20<-prediction[,1:20]

#plotting using ggplot2
library(tidyr)
```

```r
df_plot<-as.data.frame(cbind(prediction_20,time,temp))
df_long <- pivot_longer(df_plot, cols = 1:20, names_to = "Variable", values_to = "temp_Value")
library(ggplot2)
p<-ggplot(df_long) +
  geom_line(aes(x = time, y = temp_Value, color = Variable))+
  geom_point(data=df_plot,aes(x=time,y=temp))
p

#changing the parameters to get f prior regression curves agrees with your prior beliefs
#about the regression curve
#from lm set mu_0 values
beta_lm<-lm(temp~time+I(time^2),data)
coef<-beta_lm$coefficients
mu_0_opt<-c(-7.304061 ,83.149904,-78.298468)

#setting the value by seeing the plot, larger the value narrower the curve
omega_0_opt<-solve(3.5 * diag(3))
mu_opt<-3
tau_opt<-12

#setting mu and tau
prediction_opt<-joint_prior_func(50,X_mat,y,mu_opt,tau_opt,mu_0_opt,omega_0_opt)
prediction_opt_20<-prediction_opt[,1:20]

#plotting using ggplot2
library(tidyr)
df_plot_opt<-as.data.frame(cbind(prediction_opt_20,time,temp))
df_long_opt <- pivot_longer(df_plot_opt, cols = 1:20, names_to = "Variable", values_to = "temp_Value")
library(ggplot2)
p_opt<-ggplot(df_long_opt) +
  geom_line(aes(x = time, y = temp_Value, color = Variable))+
  geom_point(data=df_plot_opt,aes(x=time,y=temp))
p_opt


#1 b
#i
#calculating all the posterior parameters
beta_hat<-solve(t(X_mat)%*%X_mat)%*%t(X_mat)%*%temp
rownames(beta_hat)<-NULL

mun<-solve(t(X_mat)%*%X_mat+omega_0_opt)%*%(t(X_mat)%*%X_mat%*%beta_hat+omega_0_opt%*%mu_0_opt)
rownames(mun)<-NULL

omega_n<-t(X_mat) %*% X_mat+omega_0_opt
rownames(omega_n)<-NULL
colnames(omega_n)<-NULL

n<-nrow(X_mat)
v_n <- mu_opt+n

sigma_n<-(mu_opt*tau_opt+(t(temp)%*%temp+t(mu_0_opt)%*%omega_0_opt%*%mu_0_opt-t(mun)%*%omega_n%*% mun))/
```

```r
#draw for betas and sigma_2
post_betas<-matrix(NA, ncol = 3, nrow = 1000)
draw_invchi_vec<-rep(NA,1000)
for(i in 1:1000){
  #draw for sigma_2
  draw_invchi<-LaplacesDemon::rinvchisq(1,v_n,sigma_n)
  draw_invchi_vec[i]<-draw_invchi
  joint_prior_vec <- mvtnorm::rmvnorm(n = 1,mean=mun, sigma  =draw_invchi*solve(omega_n) )
  post_betas[i,]<-joint_prior_vec
}
#histograms of betas and sigma_2
hist(draw_invchi_vec,breaks=30,probability = TRUE)
lines(density(draw_invchi_vec),col='red')

hist(post_betas[,1],breaks=30,probability = TRUE)
lines(density(post_betas[,1]),col='red')
hist(post_betas[,2],breaks=30,probability = TRUE)
lines(density(post_betas[,2]),col='red')
hist(post_betas[,3],breaks=30,probability = TRUE)
lines(density(post_betas[,3]),col='red')


#ii

#predictions for time as columns
pre_time <- post_betas %*% t(X_mat)
get_medians<-apply(pre_time, 2, median)

#quantiles
quant<-matrix(NA, ncol = 2, nrow = length(time))
for(i in 1:length(time)){
  quant[i,]<-quantile(pre_time[,i], probs = c(0.05,0.95))
}
quant_lower<-quant[,1]
quant_upper<-quant[,2]

#plotting the temp and time
plot(time,temp)
lines(time,get_medians)
lines(time, quant_lower, col = 'red', lty = "dashed")
lines(time, quant_upper, col = 'red', lty = "dashed")

#1 c
#finding x_bar
#time with the highest expected temperature (-beta1)/(2*beta2)
calc_x_bar=(-post_betas[,2])/(2*post_betas[,3])
hist(calc_x_bar)

#1d
#see slides for answer

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

```r
Covs <- as.matrix(data_w[,2:8])
y<-as.matrix(data_w[,1])
npar<-dim(Covs)[2]
standardize <- TRUE

#prior param
mu<-as.matrix(rep(0,npar))
tau<-2
tau_2<- (tau^2)*diag(npar)

#log likelihood for optim
LogPostLogistic <- function(betas,y,Covs,mu,tau_2){
  linPred <- Covs%*%betas;
  logLik <- sum( linPred*y - log(1 + exp(linPred)) );
  #if (abs(logLik) == Inf) logLik = -20000; # Likelihood is not finite, stear the optimizer away from h
  logPrior <- dmvnorm(betas, mu, tau_2, log=TRUE);

  return(logLik + logPrior)
}

#initial beta values
initVal_betas <- matrix(0,npar,1)
OptimRes <- optim(initVal_betas,LogPostLogistic,gr=NULL,y,Covs,mu,tau_2,method=c("BFGS"),control=list(fr
#optimal betas and standard deviation
opt_betas<-OptimRes$par
std<-sqrt(diag(solve(-OptimRes$hessian)))

#posterior draws for NSmallChild
posterior_draws<-mvtnorm::rmvnorm(n = 1000, mean = opt_betas, sigma = solve(-OptimRes$hessian))[,6]
hist(posterior_draws,breaks=30)

#intervals
int<-quantile(posterior_draws,probs = c(0.025,0.975))
abline(v=int[1])
abline(v=int[2])

#verifying with glm
glm_verify<-glm(Work ~ 0 + ., data = data_w, family = binomial)


#2 b
simulating_draws<-function(x,draws){
  pred_samples<-as.vector(t(x%*%t(draws)))
  pr<-1-(exp(pred_samples)/(1+exp(pred_samples)))
  work<-ifelse(pr>0.5,'work','not work')
  get_labels<-ifelse(pr>0.5,1,0)
  return(list(pr,get_labels))
}
x<-c(1,18,11,7,40,1,1)
draws<-mvtnorm::rmvnorm(n = 1000, mean = opt_betas, sigma = solve(-OptimRes$hessian))
sim<-simulating_draws(x,draws)
hist(sim[[1]],breaks=30)
```

```
#2 c

simulating_draws<-function(x,draws,obs){
  pred_samples<-as.vector(t(x%*%t(draws)))
  pr<-1-(exp(pred_samples)/(1+exp(pred_samples)))
  bin_obs<-sapply(pr,function(x) rbinom(1,obs,x))
  return(bin_obs)
}
x <-c(1,18,11,7,40,1,1)
draws<-mvtnorm::rmvnorm(n = 1000, mean = opt_betas, sigma = solve(-OptimRes$hessian))
sim_bin<-simulating_draws(x,draws,13)
hist(sim_bin)


#question 2

library("mvtnorm")
# Load data set
WomenAtWork <- read.table("WomenAtWork.dat", header = TRUE)
#Select features and response
x<-as.matrix(WomenAtWork[2:8])
y<-as.matrix(WomenAtWork[1])
# get the number of observations and features
n<-dim(WomenAtWork)[1]
n_fea<-dim(WomenAtWork)[2]-1
# scaling factor for the prior of beta
t <- 2
# Setting up the prior mean vector and covariance matrix
mu <- as.matrix(rep(0,n_fea))
sigma <- t^2*diag(n_fea)
#return log posterior for the logistic regression
log_postlogis <- 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)
}

#initial beta
init_betas <- matrix(0,n_fea,1)

#fnscale=-1 means that we minimize the negative log posterior. Hence, we maximize the log posterior.
OptimRes <- optim(par=init_betas,
                  fn=log_postlogis,
                  gr=NULL,
                  y,x,mu,sigma,
                  method=c("BFGS"),
                  control=list(fnscale=-1),
                  hessian=TRUE)
#Naming the coefficient by covariates
Xnames<-colnames(x)
posterior_mode<-data.frame(feature_name=Xnames,beta_mode=OptimRes$par)
# Computing approximate standard deviations.
```

```r
approxpost_std <- solve(-OptimRes$hessian)
print('The posterior mode is:')
print(posterior_mode)
print('The inverse negative of the observed Hessian evaluated at the posterior mode is:')
print(approxpost_std)


#get the coefficients of the features
coef_mean<-posterior_mode[,2]
coef_covar=solve(-OptimRes$hessian)
coef<-rmvnorm(n=10000,
              mean=coef_mean,
              sigma=coef_covar)
coef_NSmallChild<-coef[,which(Xnames=='NSmallChild')]
intervals<-quantile(coef_NSmallChild,probs=c(0.025,0.975))
cat("The 95% equal tail posterior probability interval for the regression
coefficient is [",intervals[1],",",intervals[2],"]")

#compare the posterior means to the maximum likelihood estimates
glmModel<- glm(Work ~ 0 + ., data = WomenAtWork, family = binomial)
compared<-data.frame(posterior_mode,maximum_likli=glmModel$coefficients)[2:3]
print(compared)

#From the results above, by comparing the posterior means to the maximum likelihood estimates,
#we can see that the results using the maximum likelihood estimate and
#Posterior approximation are very close. Our estimation results are reasonable.

#The mean value of regression coefficient to the variable NSmallChild is -1.47248930.
#Compared to other coefficient values, the absolute value of this value is big. We can say
#that this feature is of importance for the probability that woman works. And it has big negative impac


#(b)
Constant<-1
Age<-40
NSmallChild<-1
NBigChild<-1
EducYears<-11
ExpYears<-7
HusbandInc<-18
x_new<-c(Constant,HusbandInc,EducYears,ExpYears,Age,NSmallChild,NBigChild)
simulate1<-function(x,samples,mean,sigma){
  prob<-c()
  for(i in 1:samples){
    betas<-rmvnorm(n=1,mean=mean,sigma=sigma)
    prob_y1<-exp(x_new%*%t(betas))/(1+exp(x_new%*%t(betas)))
    prob_y0<-1-prob_y1
    prob=append(prob,prob_y0)
  }
  return(prob)
}
simulate_draw1<-simulate1(x_new,1000,coef_mean,coef_covar)
hist(simulate_draw1,breaks=50,main='the posterior predictive distribution of simulate_draw1')
```

```r
#c
simulate2<-function(x,samples,mean,sigma,nsize){
  prob<-c()
  for(i in 1:samples){
    betas<-rmvnorm(n=1,mean=mean,sigma=sigma)
    prob_y1<-exp(x_new%*%t(betas))/(1+exp(x_new%*%t(betas)))
    prob_y0<-1-prob_y1
    prob=append(prob,prob_y0)
  }
  bin=rbinom(length(prob),size=nsize,prob=prob)
  return(bin)
}

simulate_draw2<-simulate2(x_new,1000,coef_mean,coef_covar,13)
hist(simulate_draw2,main='the posterior predictive distribution of simulate_draw2')
```