lab2question2

Helena Llorens Lluís (hllor282), Yi Yang (yiyan338)

2025-05-02

We aim to use a normal approximation to the posterior distribution in a Bayesian logistic regression model. The logistic regression model takes the form:

$$P(y_i = 1 \mid x_i, \beta) = \frac{1}{1 + \exp(-x_i^\top \beta)}$$

where x_i represents the covariates (including an intercept) and β are the regression coefficients.Each observation contains six variables related to a particular disease: five features (Age, Gender, Duration of Symptoms, Dyspnoea, and White Blood Count) and one intercept term.We first standardize the continuous variables — Age, Duration of Symptoms, and White Blood Count — using the formula:

$$x_{\text{standardized}} = \frac{x - \mu}{\sigma}$$

Next, we construct the log-posterior distribution of the parameters:

$$\log p(\beta \mid y, X) = \log p(y \mid \beta) + \log p(\beta)$$

where $logp(\beta)$ is the log-prior and $\log p(y \mid \beta)$ is the log-likelihood. Assuming a multivariate normal prior:

$$\beta \sim N(0, \tau^2 I)$$

where $\tau^2 = 2$.

Then we use `optim` function to find the mode of the posterior distribution (i.e., the MAP estimate) and to compute the `observed information matrix`,which is the negative of the Hessian at the mode. The approximate normal posterior is :

$$\beta \mid \mathbf{y}, \mathbf{x} \sim N\left(\tilde{\beta}, J_{\mathbf{y}}^{-1}(\tilde{\beta})\right)$$

The numeric values of $\tilde{\beta}$ and $J_y^{-1}(\tilde{\beta})$ are:

Table 1: Mode of beta

Intercept	age	gender	duration_of_symptoms	dyspnoea	white_blood
-0.3882422	-0.2661479	-0.6423896	0.19313	-0.1375241	-0.0449894

Table 2: Observed Information Matrix (Inverse Negative Hessian)

Intercept	age	gender	duration_of_symptoms	dyspnoea	white_blood
0.0977472	-0.0005446	-0.0346230	-0.0029001	-0.0797577	0.0015042
-0.0005446	0.0163897	0.0024843	-0.0001409	0.0009300	-0.0010567
-0.0346230	0.0024843	0.0631838	0.0028489	0.0027927	-0.0013382

Intercept	age	gender	duration_of_symptoms	dyspnoea	white_blood
-0.0029001	-0.0001409	0.0028489	0.0149913	0.0006546	-0.0006456
-0.0797577	0.0009300	0.0027927	0.0006546	0.0975459	-0.0006531
0.0015042	-0.0010567	-0.0013382	-0.0006456	-0.0006531	0.0163251

We then compute an approximate 95% equal-tail posterior credible interval for the variable `Age`.

```
## 95% approximate credibility intervals for Age is [ -0.5170715 -0.01522428 ].
```

Since this interval does not include `0`, we can say with 95% posterior probability (under the normal approximation) that the coefficient for Age is not zero. This implies that Age has `a statistically significant effect` on the probability of having the disease. Using `glm` function,we compare the posterior means to the maximum likelihood estimates (MLEs) obtained from the logistic regression model.

```
## [1] "Maximum Likelihood Estimates (MLE) for variable Age:"
```

```
##        age
## -0.2676216
```

```
## [1] "Posterior Means for variable Age:"
```

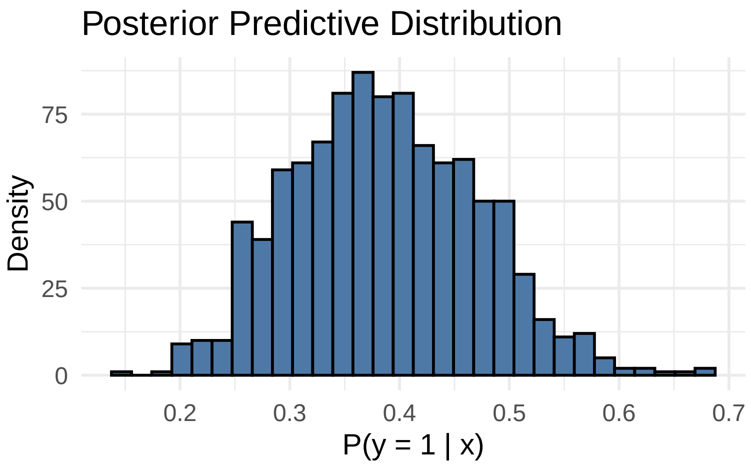
```
## [1] -0.2661479
```

After comparing, we observe that the posterior mean values closely match the MLEs from the glm Model, thereby confirming the validity and stability of our estimation procedure.

We then use the normal approximation to the posterior distribution of the regression coefficients obtained in (a), to simulate draws from the posterior predictive distribution of $Pr(y = 1 \mid x)$ for the specific individual. After standardizing the covariates, We then draw samples from the posterior distribution.For each draw of $\beta^{(s)}$,we compute:

$$Pr(y = 1 \mid x_{\text{new}}, \beta^{(s)}) = \frac{1}{1 + \exp(-x_{\text{new}}^{\top}\beta^{(s)})}$$

Finally, we visualize the distribution of predictive probabilities using a histogram.



Appendix

assignment2

```r
#assignment2
library("mvtnorm")
#(a)
data <- read.csv("Disease.csv")
data <- as.data.frame(data)
#normalize age,symtoms and white blood counts
age_mean <- mean(data$age)
age_sd <- sd(data$age)
duration_mean <- mean(data$duration_of_symptoms)
duration_sd <- sd(data$duration_of_symptoms)
white_mean <- mean(data$white_blood)
white_sd <- sd(data$white_blood)

data[,2] <- (data[,2]-age_mean)/age_sd
data[,4] <- (data[,4]-duration_mean)/duration_sd
data[,6] <- (data[,6]-white_mean)/white_sd

y <- data[,7] #response
X <- as.matrix(data[,1:6]) #Covariates
Xnames <- colnames(X)

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

  return(logLik + logPrior)
}

#set the prior
mu <- rep(0,ncol(X))
sigma <- 2*diag(ncol(X))
#initial beta
initVal <- rep(0,ncol(X))

#using optim function fine the mode of beta(mean of posterior dist) and observed information at the mod
OptimRes <- optim(initVal,LogPostLogistic,gr=NULL,y,X,mu,sigma,
                  method=c("BFGS"),control=list(fnscale=-1),hessian=TRUE)

#find the mu of approx posterior (mode of beta)
approxPostMode <-matrix(OptimRes$par,1,ncol(X))
colnames(approxPostMode) <- Xnames
#find the sd of approx posterior
approxPostStd <- sqrt(diag(solve(-OptimRes$hessian))) # Computing approximate standard deviations.
approxPostStd <- matrix(approxPostStd,1,ncol(X))
colnames(approxPostStd) <- Xnames
```

```
#find the 95% equal tail posterior probability interval for coefficient Age
Cred_int <- matrix(0,2,ncol(X)) # Create 95 % approximate credibility intervals for each coefficient
Cred_int[1,] <- approxPostMode - 1.96*approxPostStd
Cred_int[2,] <- approxPostMode + 1.96*approxPostStd
colnames(Cred_int) <- Xnames

cat("95% approximate credibility intervals for Age is [",Cred_int[1,2],Cred_int[2,2],"].","\n")

#Comparison with glm
glmModel<- glm(class_of_diagnosis ~ 0 + ., data = data, family = binomial)
print("Maximum Likelihood Estimates (MLE) for variable Age:")
coef(glmModel)[2]
print("Posterior Means for variable Age:")
approxPostMode[2]

#(b)
#normalizing the sample
xnew <- c(1,38,1,10,0,11000)
xnew_normalized <- c(1,(38-age_mean)/age_sd,1,(10-duration_mean)/duration_sd,
                     0,(11000-white_mean)/white_sd)

#sampling beta from the posterior distribution
set.seed(12345)
beta_sample <- rmvnorm(1000,mean =approxPostMode,sigma=solve(-OptimRes$hessian))

#compute the predictive probability for different beta samples
predict_probs <- numeric(nrow(beta_sample))
for (i in 1:nrow(beta_sample)) {
  beta <- beta_sample[i,]
  linPred <- sum(xnew*beta)
  prob <- 1/(1+exp(-linPred))
  predict_probs[i] <- prob
}


hist(predict_probs,breaks = 30,probability = T,main = "Posterior Predictive Distribution",xlab = "P(y=1
```