Section 3

Vera Liu

April 6, 2018

1

Set $\tau^2 = 1$ as fixed, the conditionals would be expressed as following:

$P(\beta|x,z) \sim N((X^TX+I)^{-1}X^TZ, (X^TX+I)^{-1})$, this is the same as previous exercise, since the dependency of β on observed data is all through the latent variable Z, which takes continuous numeric values and is the same as usual linear regression setup.

$P(z|\beta,x,y) \sim TN(X^T\beta, 1)$, where TN stands for a truncated normal, specifically

$$Z \begin{cases} \sim N(X^t\beta,1) \text{ and } Z > 0 & \text{if } y = 1 \\ \sim N(X^t\beta,1) \text{ and } Z < 0 & \text{if } y = 0 \end{cases}$$

Read in data:

```
library(readr)
pima<- read_csv("~/pima-indians-diabetes.csv")
xmat<-as.matrix(pima[,1:8])
ymat<-as.matrix(pima[,9])
```

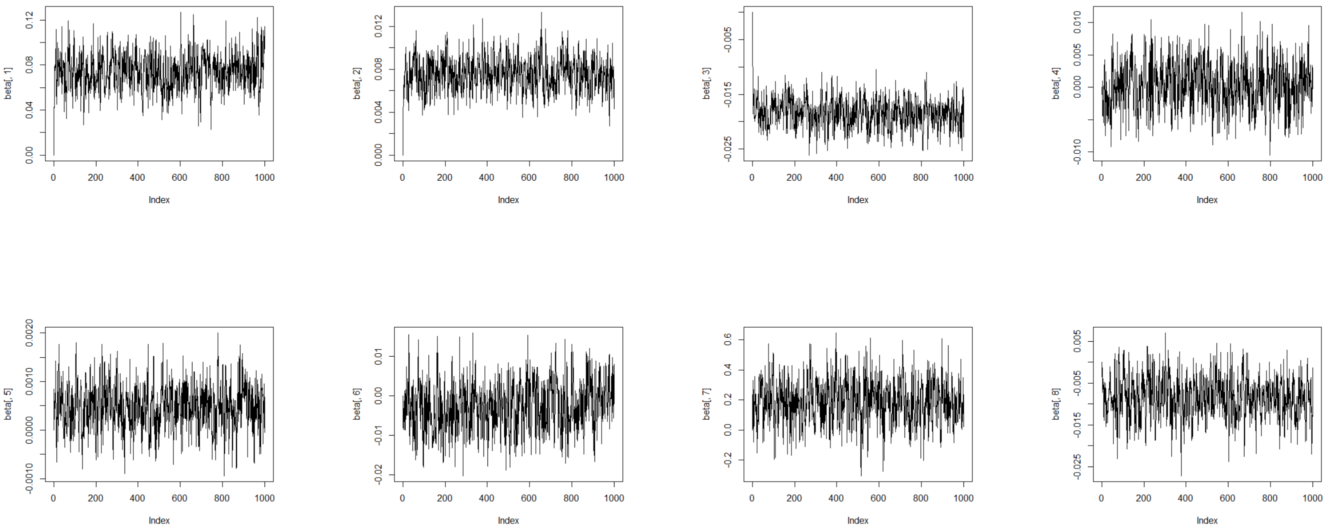
Set up the vectors of interests:

```
tau<-1
n<-1000
beta<-matrix(rep(NA,(n+1)*8), n+1,8)
z<-matrix(rep(NA,n*nrow(pima)), n, nrow(pima))
beta[1,]<-rep(0,8)
```

Implementation of Gibbs Sampler:

```
library(truncnorm)
for (i in 1:n){
  for (j in 1:nrow(pima)){
    if (ymat[j] == 1){
z[i,j] <- rtruncnorm(1, a=0, b=Inf, mean=(t(xmat[j,]) %*% beta[i,]), sd=1)}
    else {z[i,j] <- rtruncnorm(1, a=-Inf, b=0, mean=(t(xmat[j,]) %*% beta[i,]), sd
        =1)}
    }
beta[i+1,]<-mvrnorm(1, solve(t(xmat) %*% xmat +diag(dim(xmat)[2])) %*% (t(xmat) %*%
    z[i,]), solve(t(xmat) %*% xmat+diag(dim(xmat)[2])))
}
```

Following is the trace plot for parameter estimates, where all of them have good mixtures.



2

With $y \sim Bern(\frac{1}{1+e^{-x\beta}})$

$f(\beta|y) \propto f(\beta)f(y|\beta) \propto e^{-\frac{1}{2}\beta^2} \prod_i (\frac{1}{1+exp(-x_i\beta)})^{y_i}((1 - \frac{1}{1+exp(-x_i\beta))})^{1-y_i}$

The target of minimization could be the corresponding negative log-likelihood, which is $\frac{1}{2}\beta^2 + \sum_i y_i(log(\frac{1}{1+exp(-x_i\beta)})) + (1-y_i)(log\frac{exp(-x_i\beta)}{1+exp(-x_i\beta)})$

Read in data:

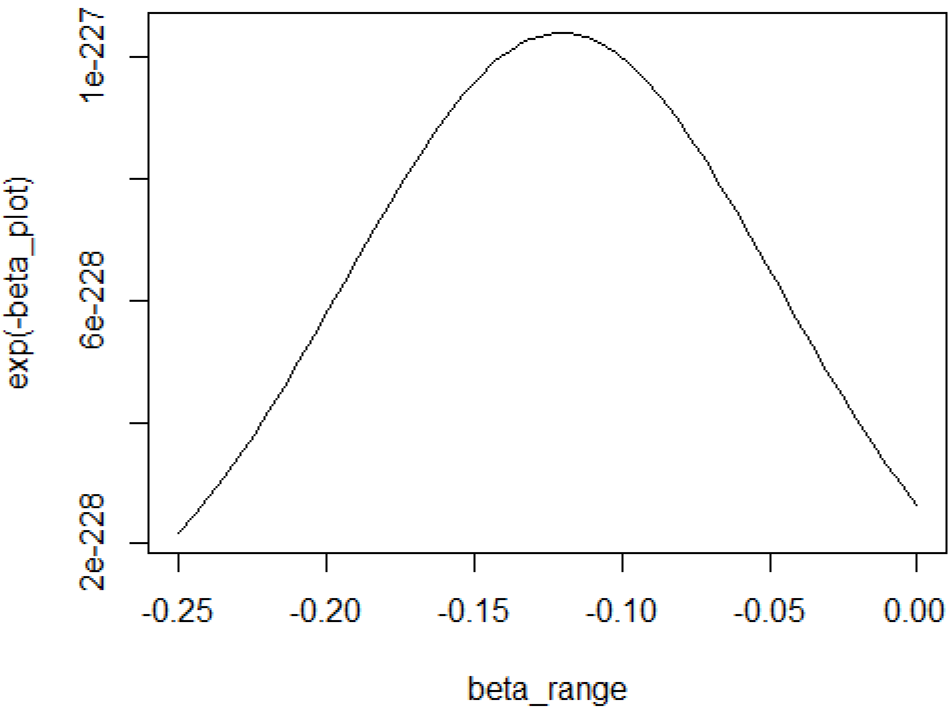
```
titanic <- read_csv("~/titanic.csv.txt")
titanic<-titanic[complete.cases(titanic), ]
titanic$Survived<-as.numeric(titanic$Survived == "Yes")
titanic$Age<-scale(titanic$Age)
```

Set up the function of with respect β, to be optimized.

```
agefunc<-function (x,y,beta) {
  summation<-0
  for (i in 1:length(x)){
    summation <- summation + y[i]*log(1/(1+exp(-x[i]*beta))) + (1-y[i]) * log(exp(-x
        [i]*beta)/(1+exp(-x[i]*beta)))
  }
  logneg<-0.5*beta^2 - summation
  return(logneg)
}
```

The optimize function in R is used to do this one-dimensional optimization.

```
beta_map<-optimize(agefunc, c(-1, 1), tol = 0.0001, x=titanic$Age, y=
    titanic$Survived)$minimum
```



Get $\hat{\beta}_{MAP} = -0.1205651$

Plot the density of β for $\beta \in (-0.25, 0)$, and confirm that the β estimate around -0.12 is indeed reasonable.

```
n=50
beta_plot<-rep(NA, n)
beta_range<-seq(from = -0.25, to = 0, length.out = n)
for (i in 1:n){
  beta_plot[i] <-agefunc(titanic$Age, titanic$Survived, beta_range[i])
}
plot(beta_range,exp(-beta_plot), type="l")
```

3

Using Laplace approximation, $Q^*(x) = P^*(x^*)exp(-\frac{c}{2}(x-x^*)^2)$, the mean for this unnormalized Gaussian pdf is $x^* = \hat{\beta}_{MAP}$, and precision is $c = -\frac{\partial^2}{\partial x^2} logP^*(x)|_{x=x^*}$

$-\frac{\partial^2}{\partial x^2} logP^*(\beta) = 1 + \sum_i (x_i)^2 exp\{-\beta x_i\} \frac{1}{(1+exp\{-\beta x_i\})^2}$, then plug in the x_i's

```
c<-1
for (i in 1:length(titanic$Age)){
  c<- c + titanic$Age[i]^2*exp(-beta_map*titanic$Age[i])/(1+exp(-beta_map*
      titanic$Age[i]))^2
```

```
}
```

The precision estimation is $c = 187.74$, and mean estimation is the same as $\hat{\beta}_{MAP}$

4

Manually create an intercept column

```
titanic$Intercept <-1
```

The optimization to find maximum posteriori estimate is now through a vector form:

```
multifunc<-function (beta,x,y) {
  summation<-0
  for (i in 1:dim(x)[1]){
    summation <- summation + y[i]*log(1/(1+exp(-t(x[i,]) %*% beta))) + (1-y[i]) *
        log(exp(-t(x[i,]) %*% beta)/(1+exp(-t(x[i,]) %*% beta)))
  }
  logneg<- 0.5* t(beta) %*% beta - summation
  return(logneg)
}

xmulti<-as.matrix(titanic[,c(3,6)])
optim(c(-1,-1),multifunc, x=xmulti, y=titanic$Survived)
```

$\hat{\beta}_{Inter_MAP} = -0.3470527$ and $\hat{\beta}_{Age_MAP} = -0.1246329$
Here the precision c is expressed as the Hessian matrix evaluated at the point estimates.

```
library(numDeriv)

hessfun<-function(beta){
   0.5* t(beta) %*% beta - t(as.matrix(titanic$Survived)) %*% log(1/(1+exp(-xmulti
      %*% beta))) - t(1-(as.matrix(titanic$Survived))) %*% log(1-1/(1+exp(-xmulti
      %*% beta)))
  }

precmatri<-hessian(hessfun,c(-0.1246329,-0.3470527))

sqrt(1/precmatri[1,1])
sqrt(1/precmatri[2,2])
```

The 95 percent credible interval for age coefficient is $(-0.27036464, 0.02109884)$ and for intercept is $(-0.4916306, -0.2024748)$.

5

A Poisson model might not be appropriate because:

- The minimum value of the data is 5. This doesn't match with the regular Poisson distribution, where the support should be all non-negative integer values.
- The mean of the data is around 16 and variance aroumd 460, which doesn't match with a poisson model that these two values should be equal.

6

Same as optimizing the posterior before, to estimate grade coefficient and intercept,

```
multifunc_censor<-function (beta,x,y)
  {
  0.5* t(beta) %*% beta - t(as.matrix(y)) %*% x %*% beta + matrix(rep(1,dim(x)[1]),
      nrow=1) %*% exp(x%*%beta)
  }

xmulti_grade<-as.matrix(censor[,c(3,9)])
optim(c(-1,-1),multifunc_censor, x=xmulti_grade, y=censor$ACTIONS)
```

The maximum posteriori estimate is $\hat{\beta} = 0.05031772$, $Intercept = 2.38863366$

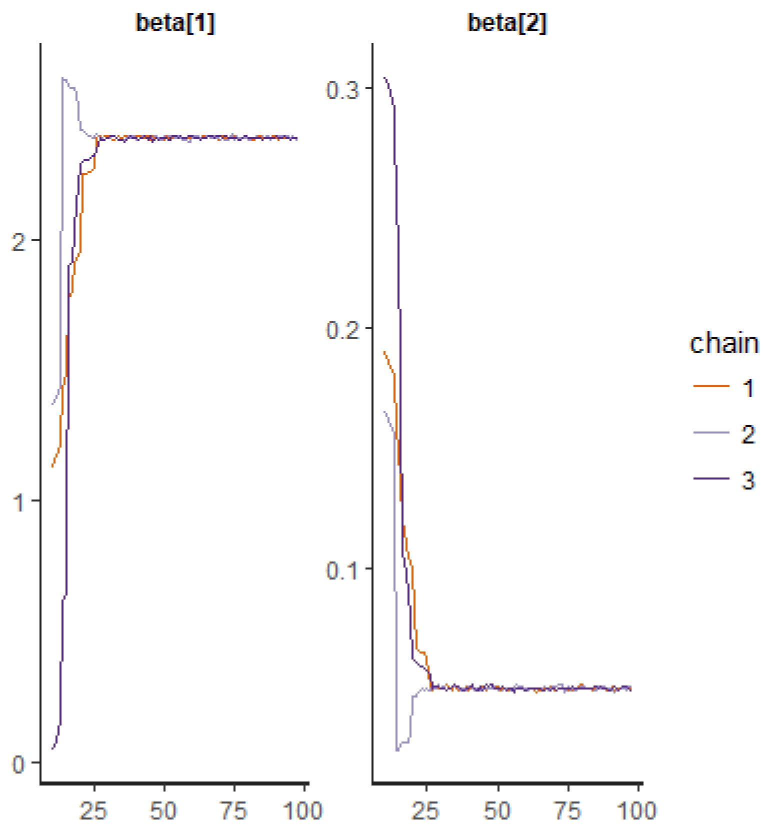
The Hessian matrix gives the precision matrix estimation:

```
hessfun2<-function(beta){
  0.5* t(beta) %*% beta - t(as.matrix(censor$ACTIONS)) %*% xmulti_grade %*% beta +
      matrix(rep(1,dim(xmulti_grade)[1]),nrow=1) %*% exp(xmulti_grade%*%beta)
}

precmatri2<-hessian(hessfun,c(0.05031772,2.38863366))

sqrt(1/precmatri2[1,1])
sqrt(1/precmatri2[2,2])
```

The 95 percent credible interval for grade coefficient is $(-0.2062942, 0.3069142)$ and for intercept is $(2.133973, 2.643294)$.

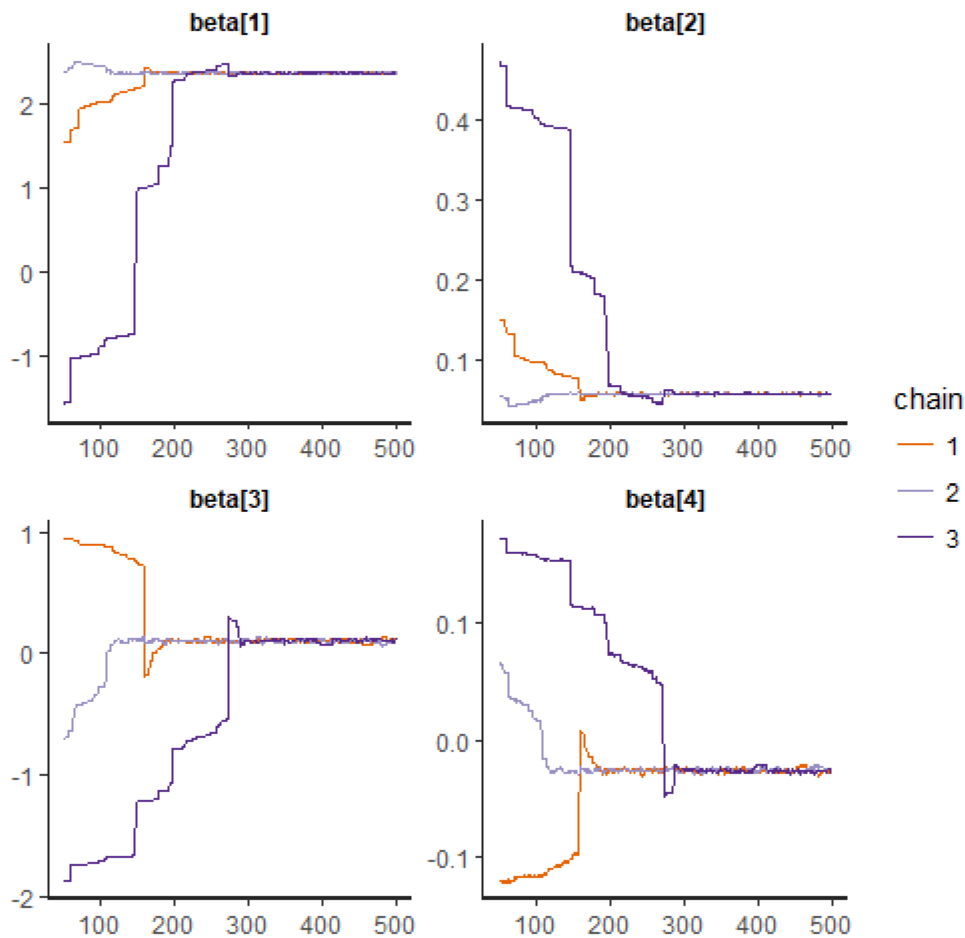
7

Attached is the trace plot from stan MCMC sampling. Here β_1 stands for intercept, and β_2 stands for age coefficient. It's no surprise that they maintain similar values as from our Laplace approximation.



8

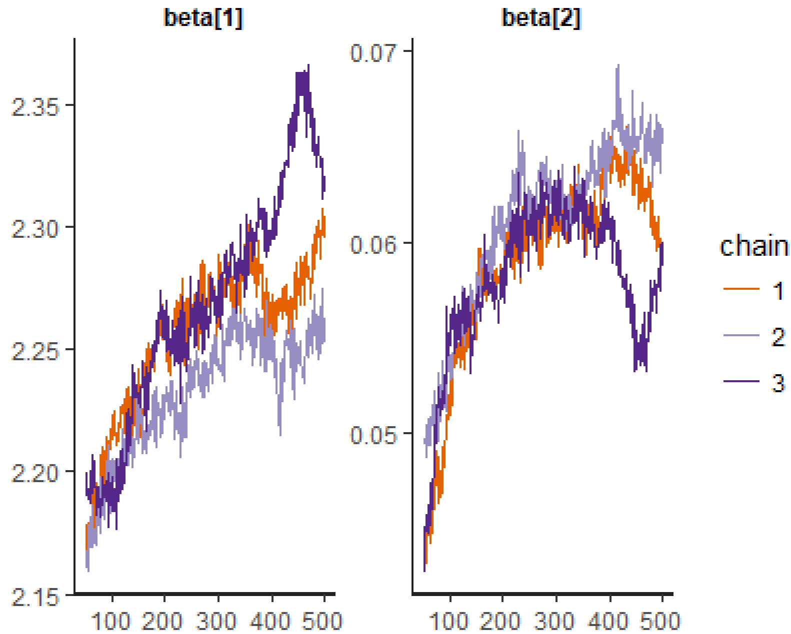
To modify the model a bit, first the sec variable and sex-grade interaction is added. This results in two new coefficients to estimate (β_3 = sex, and β_3 = sex-grade interaction)



As pointed out before, Poisson model is not a perfectly realistic choice here, in the sense that the over-dispersion of data is not taken into consideration. Hence, another parameter ϵ will be included to increase the variance of the model, thus giving it more flexibility, since we have the constraint that mean and variance have to be approximately equal in a real Poisson model.

Before, the model is $y \sim Pois(e^{x\beta})$, now with the new parameter, it becomes $y \sim Pois(e^{x\beta}\epsilon)$, where ϵ is distinctive for each y, and $\epsilon \sim Gamma(1, 1)$. Notice that the mean of y is not changed by multiplying ϵ, but the variance is increased.

Following shows the new trace plot for intercept and age coefficient under this model with the extra parameter.



9

We can introduce another latent variable z, denoting whether observation y is censored or not.

This will make the full model as follows:

$y_i = z_i 1_{z_i \geq 5} - 991_{z_i < 5}$

$z_i \sim Pois(\lambda_i)$

$\lambda_i = x_i \beta$

The model that take care of the censoring could then be implemented.