

Assignment 3.b for **STATGR6103**

submitted to Professor Andrew Gelman

Advait Rajagopal

26 September 2016

1 Question 1

Consider the following Poisson regression model;

$$y_i \sim \text{Poisson}(u_i * \exp(\alpha + \beta * x_i)) \quad (1)$$

Set $n = 10$, x to the values 1, 2, . . . , 10, u to the values created by `rep(c(0.1,0.5),c(5,5))`, and set (α, β) to the true values of (1.0, 0.3).

1.1 Part A

Simulate data y from the model and make a scatterplot of y_i/u_i vs. x_i .

Figure 1 shows the estimates of y_i/u_i when the regression model in equation 1 is used to simulate the data y and the ratio y_i/u_i is plotted versus the x_i values.

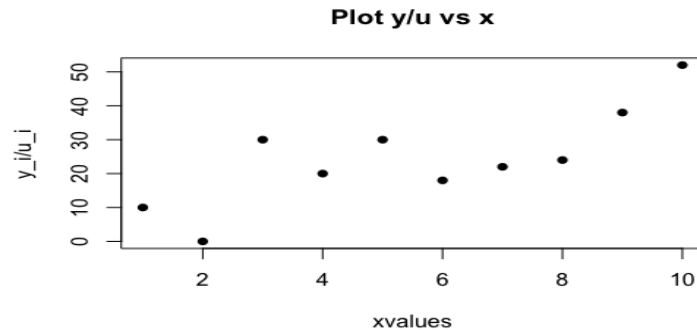


Figure 1: y/u values for x values

1.2 Part B

Now forget the true values of α, β . Write the posterior density for (α, β) , assuming a noninformative uniform prior distribution. We know the likelihood distribution is given as;

$$p(y_i | \alpha, \beta, u_i, x_i) \sim \text{Poisson}(u_i * \exp(\alpha + \beta * x_i)) \quad (2)$$

Since (α, β) follow a noninformative uniform prior this is paramount to saying¹;

$$p(\alpha, \beta) \propto 1 \quad (3)$$

¹page 75, Gelman's BDA 3

From equations 2 and 3 we are able to derive the posterior distribution of the parameters α and β as follows because the posterior distribution is proportional to just the likelihood as we have an uninformative prior;

$$p(\alpha, \beta | y, x, u) \propto \prod_{i=1}^{10} \frac{\exp(-u_i \exp(\alpha + \beta x_i)) * (u_i \exp(\alpha + \beta x_i))^{y_i}}{y_i!} \quad (4)$$

1.3 Part C

Using R, compute the posterior mode and the covariance matrix around the mode. Make a plot showing the mode and an ellipse showing the normal approximation. Also on this plot indicate the true value of (α, β)

For this part I use the “LearnBayes” and “mvtnorm” packages in R. The *laplace* function which is part of the “LearnBayes” package allows me to summarize a posterior density function and computes the mode of the parameters α and β as well as estimates the associated variance matrix, which is the inverse of the *observed information* and is a positive definite matrix². The mode of (α, β) is $[1.1310138, 0.2877877]$. The variance matrix is;

$$I(\theta)^{-1} = \begin{bmatrix} 0.28490902 & -0.0328403 \\ -0.0328403 & 0.003942627 \end{bmatrix}$$

The “mvtnorm” package allows you to estimate the density of a function using *dmvnorm* for the multivariate normal distribution with a specified mean (where we use the modal values obtained from *laplace* function) and the aforementioned covariance matrix³. Figure 2 shows the mode (α, β) as obtained by normal approximation and the true values of the same that we had forgotten in Part B.

Contours with normal approximations of alpha and beta

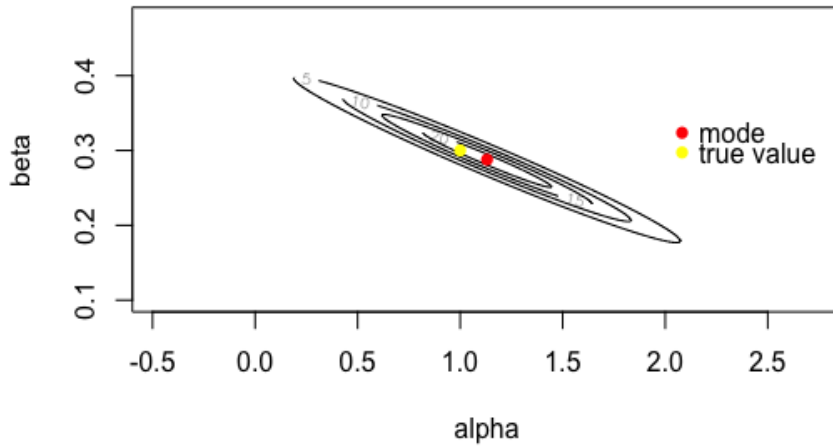


Figure 2: Mode in “Red” and the true value in “Yellow”

²page 84, eqn 4.2 Gelman’s BDA 3

³For a full exposition see the Mathematical Appendix in Section 2

1.4 Part D

Fit the model in Stan and then plot random draws from the joint posterior distribution of (α, β) by drawing from a grid, adding these dots to the graph above. For this part of the model I fit the regression equation described in equation 1 with the the likelihood function described in equation 2. The prior distribution of the underlying parameters α and β is a uniform distribution. The posterior estimates of the parameters are given below;

Table 1: Posterior estimates of parameters

	mean	2.5%	25%	50%	75%	97.5%
α	1.05	-0.11	0.71	1.09	1.44	2.01
β	0.30	0.18	0.25	0.29	0.34	0.43

Figure 3 shows the posterior draws of α and β from the posterior distribution of α and β as estimated by Stan overlaying the contour plot in Figure 2.

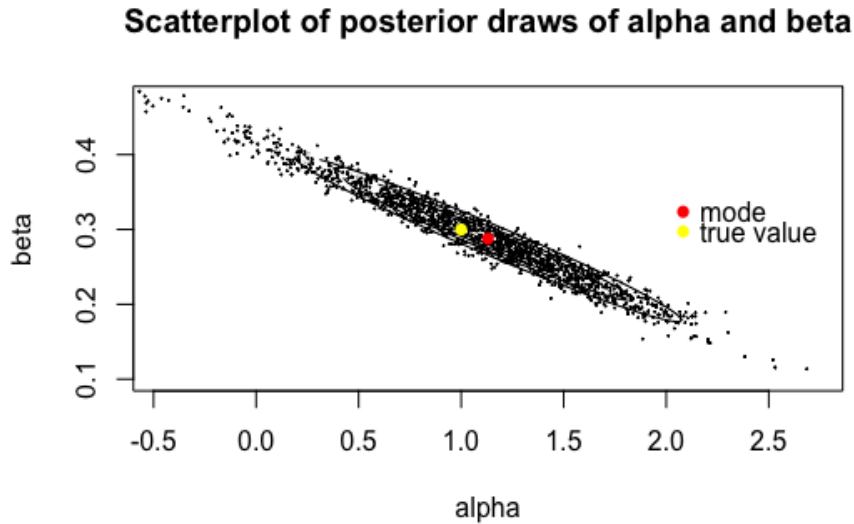


Figure 3: Scatterplot of posterior draws of α and β

We observe that Stan fits the model wonderfully close to the true values of α and β which it didn't know. A few other times I ran the model with different seeds I got some varying results but I have presented the results of one particularly close estimation that I achieved with no `set.seed()`.

2 Mathematical Appendix

I thought it would be beneficial to explain how the inverse of the covariance matrix is actually derived from the posterior distribution of α and β . We begin by considering equation 4;

$$p(\alpha, \beta | y, x, u) \propto \prod_{i=1}^{10} \frac{\exp(-u_i \exp(\alpha + \beta x_i)) * (u_i \exp(\alpha + \beta x_i))^{y_i}}{y_i!}$$

Taking log on both sides;

$$\begin{aligned} \log(p(\alpha, \beta | y, x, u)) &= \log\left(\prod_{i=1}^{10} \frac{\exp(-u_i \exp(\alpha + \beta x_i)) * (u_i \exp(\alpha + \beta x_i))^{y_i}}{y_i!}\right) \\ &= \sum_{i=1}^{10} -u_i \exp(\alpha + \beta x_i) + y_i [\log(u_i) + \alpha + \beta x_i] - \log(y_i!) \\ &= \sum_{i=1}^{10} y_i \log(u_i) + y_i \log(\alpha) + y_i \log(\beta x_i) - u_i \exp(\alpha + \beta x_i) - \log(y_i!) \end{aligned}$$

Now we take the partial derivatives of the simplified logarithmic form of the posterior distribution. The second order own partial derivatives make the diagonal elements and the symmetric second order cross partial derivatives make up the off diagonal elements.

$$\frac{\partial \log p(\alpha, \beta | y, x, u)}{\partial \alpha} = \sum_{i=1}^{10} y_i - u_i \exp(\alpha + \beta x_i)$$

$$\frac{\partial^2 \log p(\alpha, \beta | y, x, u)}{\partial \alpha^2} = \sum_{i=1}^{10} -u_i \exp(\alpha + \beta x_i) \quad (5)$$

$$\frac{\partial \log p(\alpha, \beta | y, x, u)}{\partial \beta} = \sum_{i=1}^{10} y_i x_i - u_i x_i \exp(\alpha + \beta x_i)$$

$$\frac{\partial^2 \log p(\alpha, \beta | y, x, u)}{\partial \beta^2} = \sum_{i=1}^{10} -x_i^2 u_i \exp(\alpha + \beta x_i) \quad (6)$$

$$\frac{\partial^2 \log p(\alpha, \beta | y, x, u)}{\partial \alpha \partial \beta} = \frac{\partial^2 \log p(\alpha, \beta | y, x, u)}{\partial \beta \partial \alpha} = \sum_{i=1}^{10} -x_i u_i \exp(\alpha + \beta x_i) \quad (7)$$

Equations 5 and 6 show the own second order partial derivatives and equation 7 is the cross partial second order derivative. These elements form the covariance matrix described earlier.

3 Code

3.1 R Code

Listing 1: R Code

```
setwd("/Users/Advait/Desktop/New_School/Fall16/BDA/Class5")
n <- 10
x <- c(1:10)
u <- rep(c(0.1,0.5),c(5,5))
atrue <- 1
btrue <- 0.3
y <- rpois(10, u*(exp(atrue + btrue*x)))
#Plot y/u
plot(x, (y/u), pch = 16, xlab = "xvalues", ylab = "y_i/u_i",
main = "Plot_y/u_vs_x")
##Stan model
library(rstan)
stanc("3b.stan")
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
fit1 <- stan("3b.stan", data = list("y","x","n","u"),
            iter = 1000, chains = 3)

print(fit1)
ext1 <- extract(fit1)
atrial <- ext1$alpha
btrial <- ext1$beta

##approximation
install.packages("LearnBayes",dependencies = T)
library(LearnBayes)
install.packages("mvtnorm", dependencies = T)
library(mvtnorm)
##Finding mode and covariance matrix
logpost <-function(theta){
  pf = 0
  a<-theta[1]
  b<-theta[2]
  for (i in (1:length(x))){
    pf2 = (a+b*x[i])
    pf = pf + y[i]*(log(u[i]) + pf2) - u[i]*exp(pf2) -
          log(factorial(y[i]))
  }
  pf}
summary <- laplace(logpost,c(1,0.3))
```

Listing 2: R Code contd.

```

alphanew <- seq(summary$mode[1] -
                3*sqrt((summary$var[1,1])),
                summary$mode[1] +
                3*sqrt((summary$var[1,1])),
                length.out = 500)
betanew <- seq(summary$mode[2] -
                3*sqrt((summary$var[2,2])),
                summary$mode[2] +
                3*sqrt((summary$var[2,2])),
                length.out = 500)

postmod <- c(summary$mode)
cov_matrix <- summary$var
zmatrix <- matrix(0,500,500)
length(postmod)
length(cov_matrix)
for (i in 1:500){
  for(j in 1:500){
    zmatrix[i,j] = dmvnorm(c(alphanew[i], betanew[j]),
                           postmod, cov_matrix)}}

#plot1
contour(alphanew, betanew, zmatrix, nlevels = 8,
        xlab = "alpha", ylab = "beta",
        main = "Contours_with_normal_approximations_of_alpha_and_beta")
points(1,0.3, pch = 16, cex = 1, col = "yellow")
points(postmod[1],postmod[2],pch = 16, cex = 1, col = "red" )
legend(2,.35,legend=c("mode","true_value"),
col=c("red","yellow"),cex=1,pch=c(16),bty="n")

#plot2
contour(alphanew, betanew, zmatrix, nlevels = 8,
        xlab = "alpha", ylab = "beta",
        main = "Scatterplot_of_posterior_draws_of_alpha_and_beta")
for (i in 1:1500){
  points(atrial[i], btrial[i], cex = .1)
}
points(1,0.3, pch = 16, cex = 1, col = "yellow")
points(postmod[1],postmod[2],pch = 16, cex = 1, col = "red" )
legend(2,.35,legend=c("mode","true_value"),
        col=c("red","yellow"),cex=1,pch=c(16),bty="n")

```

Listing 3: Stan Code

```
data{
  int n;
  int y[n];
  real x[n];
  real u[n];
}
parameters{
  real alpha;
  real beta;
}
transformed parameters{
  real lambda[n];
  for (i in 1:n)
    lambda[i] = u[i]*(exp(alpha + beta*(x[i])));
}
model{
  y ~ poisson(lambda);
}
```