Logit Regression

Javier Esteban Aragoneses, Mauricio Marcos Fajgenbaun, Danyu Zhang, Daniel Alonso

March 18th, 2021

Introduction

In this Project, we intend to build and study a model to explain weather a costumer will do a certain purchase or not. In order to do this, we will try to explain this decision (either purchasing or not purchasing) by the number of bets that the person does in a well-known website. With this, we may be able to predict if a person will buy or not, depending on the numer of bets already mentioned.

Model

In general, in Bayesian Logistic Regresion, you start with an initial belief about the distirbution of $p(\alpha, \beta)p(\alpha, \beta)$.

Then $p(\alpha, \beta x, y) \propto p(y\alpha, \beta, x)p(\alpha, \beta)p(\alpha, \beta x, y) \propto p(y\alpha, \beta, x)p(\alpha, \beta)$. That is, the posterior, which is our updated belief about the weights given evidence, is proportional to our prior (initial belief) times the likelihood. We can't evaluate the closed form posterior, bit can approximate it by sampling or variational methods. This gives us a distribution over the weights. Finally, we will get a distribution for the parameters and we will study the mean, variance, median and credible intervals.

Our dependent variable "Y" is a random variable that distributes as a Bernoulli (it can take either value "1" if the costumer buys and "0" if the costumer does not buy). We will build a regression model, using as an independent variable "number of bets", that is a quantitative discrete variable.

$$Y_i|p_i \overset{ind}{\sim} \text{Bernoulli}(p_i),\ i = i, \ldots, n.$$

The logistic regression model writes that the logit of the probability p_i is a linear function of the predictor variable x_i:

$$logit(p_i) = log(\frac{p_i}{1-p_i}) = \alpha + \beta X_i$$

By rearranging the logistic regression equation, we can express it as a nonlinear equation of the probability of success p_i:

$$p_i = \frac{e^{\alpha+\beta X_i}}{1 + e^{\alpha+\beta X_i}}$$

With this formula we can guarantee that the probability will lie in the interval [0,1].

As for any Bayesian inference exercise, we have 3 main parts that we can never avoid:

- Write the likelihood of the data
- Form a prior distribution over all unknown parameters
- Use Bayes theorem to find the posterior distribution over all parameters

Likelihood Function

Here, the likelihood contribution from the ith subject is binomial:

$$\text{likelihood}_i = p(x_i)^{y_i}(1-p(x_i))^{(1-y_i)}$$

Where $p(x_i)$ represents the probability of the event for subject I, which has covariate x_i and y_i indicates the presence, $y_i = 1$ or absence $y_i = 0$ of the purchase for that subject. Of course, in logistic regression, we know that

$$p(x) = \frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}}$$

Rewritten as:

$$\text{likelihood}_i = (\frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}})^{(y_i)}(1-\frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}})^{(1-y_i)}$$

And as the individual subjects are assumed independent from each other, the likelihood function over a data set of n subjects is:

$$\text{likelihood} = \prod_{i=1}^{n}[(\frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}})^{y_i}(1-\frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}})^{(1-y_i)}$$

Prior Distribution

The set of unknown parameters are "alfa" and "beta". In general, any prior distribution can be used, depending on the available prior information. Here, we will use two "common" priors (given by the professor):

$$\alpha \sim N(0,5)$$
$$\beta \sim N(0,2)$$

Posterior distribution

The posterior distribution is derived by multiplying the prior distribution over all parameters by the full likelihood function, so that:

$$\text{posterior} = \prod_{i=1}^{n}[(\frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}})^{y_i}(1-\frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}})^{(1-y_i)}] \times \prod_{j=0}^{p}\frac{1}{\sqrt{2\pi}\sigma_j}e^{-\frac{1}{2}(\frac{\beta_j-\mu_j}{\sigma_j})^2}$$

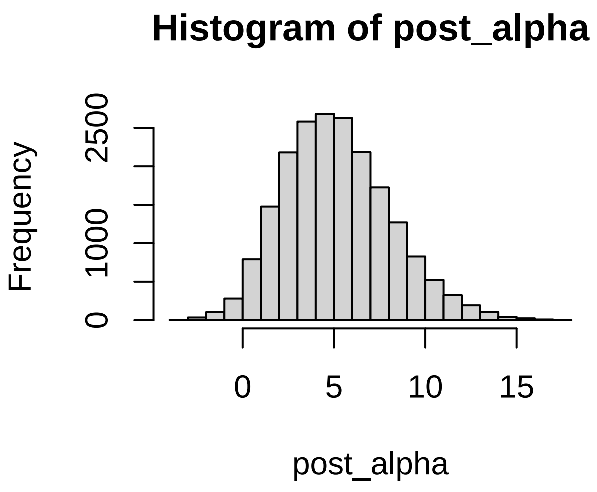
Where u_j and $sigma_j$ are the already expressed mean and variance of the normal distribution for "alpha" and "beta". Of course, the above expression has no closed form expression, and even if it did, we would have to perform multiple integration to obtain the marginal distribution for each coefficient. So we will use the Gibbs sampler as implemented in Stan, to solve approximate the properties of the marginal posterior distribution for "alpha" and "beta".

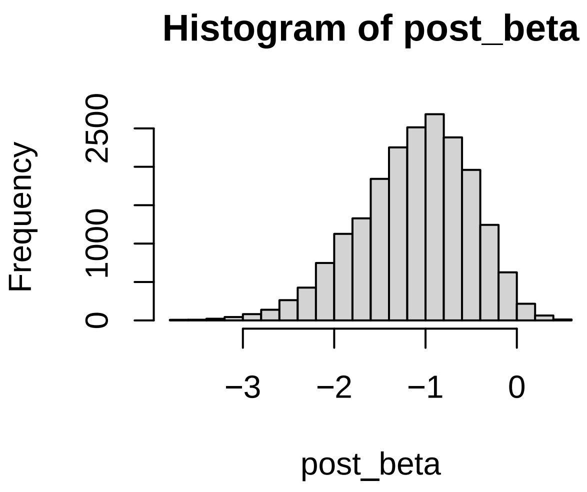
We can see that in 10000 iterations, only approximately 3300 are independent. However, all R are close to 1, in order words, if we try with different chains, the posterior probabilities will converge in the same way. So, we can trust the simulation.

Solution of the real problem

Posterior α and β distribution

Given prior knowledge for α and β (our coefficient parameters of the logistic regression) and the statistical models we used, the probability distribution of α (our intercept) and β (our coefficient) are:



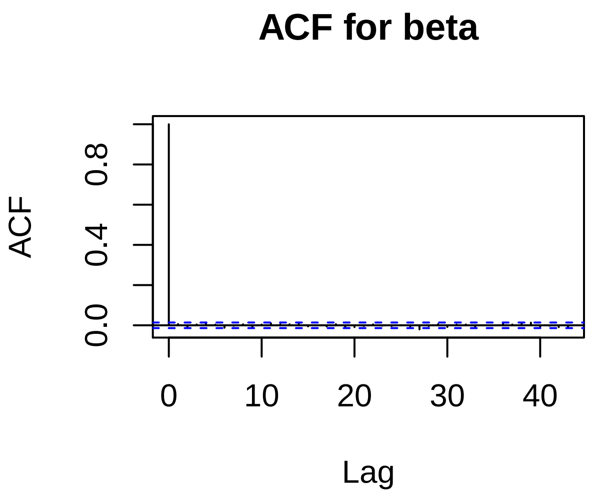


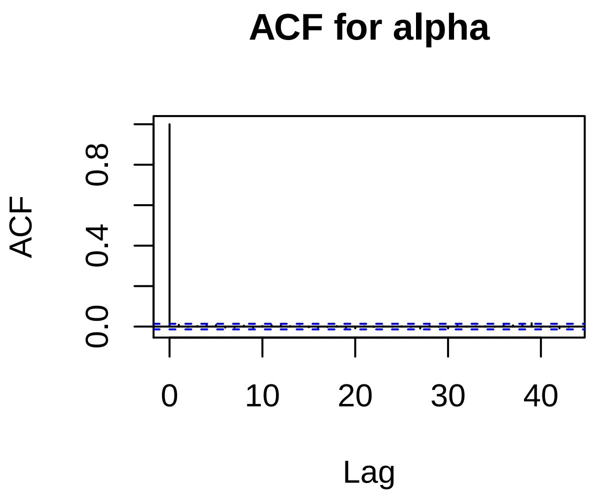
So beta has a distribution with mean equal to -1.09 and a standard error of 0.01. The mean of beta is with 95% of probability between:

```
#>      2.5%      97.5% 
#>  0.1441783 11.6084287
```

So beta has a distribution with mean equal to -1.09 and a standard error of 0.01. The mean of beta is with 95% of probability between:

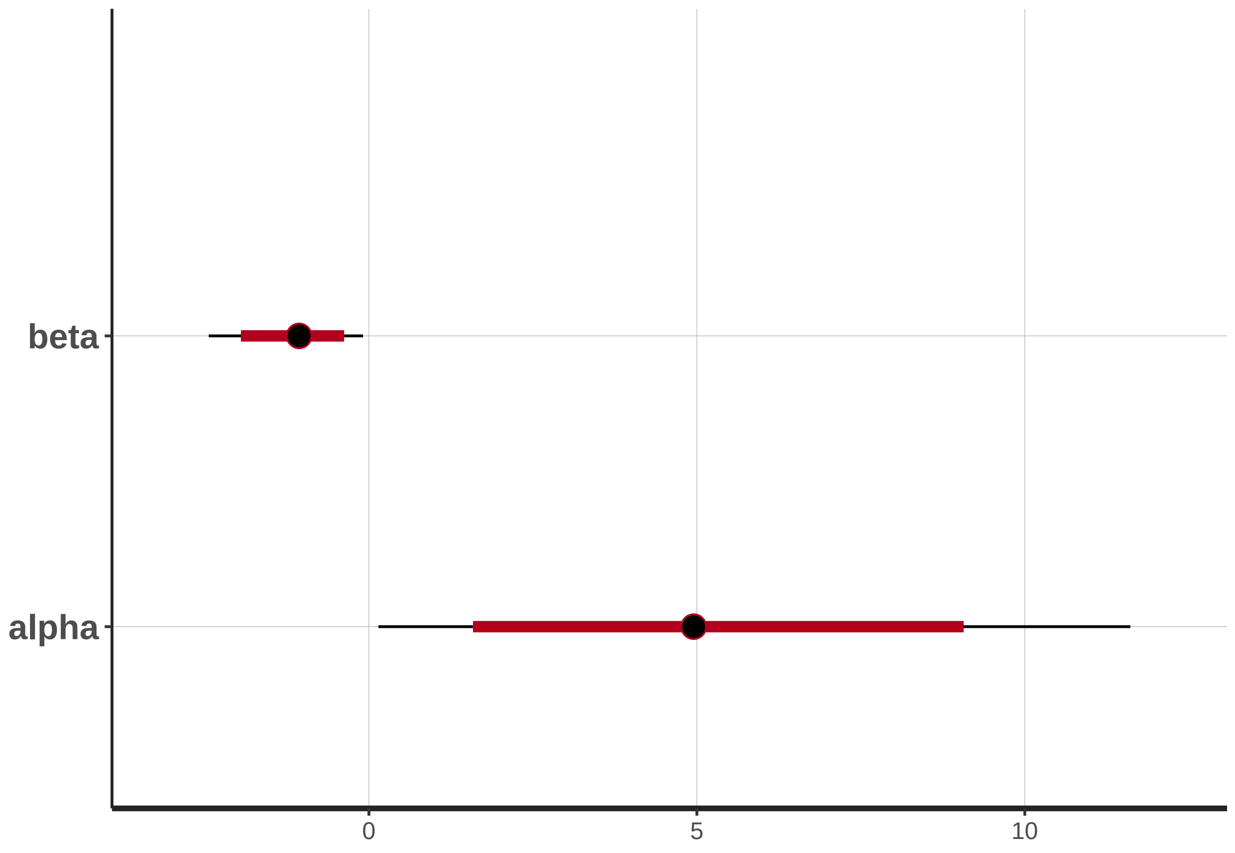
```
#>        2.5%       97.5% 
#> -2.44220175 -0.08974735
```





Once the simulated values are found, one applies several diagnostic procedures to check if the simulation appears to converge to the posterior distribution.

From viewing these plots (for the regression parameters α and β), it appears that there is a small amount of autocorrelation in the simulated draws and that the draws appear to have converged to the posterior distribution.



Annex: code

```
# Importing libraries
library(dplyr)
library(rstan)
library(shinystan)
library(coda)

# STAN model setup
m1 <- "
data {
int N;                             # number of observations
int purchase[N];                   # setting the dependent variable as binary
vector[N] x;                       # independent variable 1
}
parameters {
real alpha;                        # intercept
real beta;                         # beta for educate, etc
}
model {
alpha ~ normal(0,5);               # you can set priors for all betas
beta ~ normal(0,2);
purchase ~ bernoulli_logit(alpha + beta * x);    # model
}
generated quantities {
  vector[N] y_pred;
  y_pred <- beta * x; //the y values predicted by the model
}
```

```r
"

# Dataset setup
purchase <- c(1,0,1,1,0,0)
x <- c(3,5,2,5,10,6) # number of bets
df <- list(N = length(x), purchase = purchase, x = x) # dataset setup

# STAN fit
fit <- stan(model_code = m1, data = df, pars = c("beta","alpha","y_pred"), iter = 1e4)

# plotting posterior distribution of beta
post_beta <- extract(fit, pars="beta")
plot(post_beta)

# plotting posterior distribution of alpha
post_alpha <- extract(fit, pars="alpha")
plot(post_alpha)

# Autocorrelation plots for beta and alpha
acf(extract(fit, 'beta')$beta)
acf(extract(fit, 'alpha')$alpha)

# plotting beta and alpha
plot(fit,pars=c("beta","alpha"))
```

Annex: summary stats for the fit

Our summary statistics for the model are:

```
#> Inference for Stan model: ff8b94d4d8110c09dbe34c20328c9788.
#> 4 chains, each with iter=10000; warmup=5000; thin=1;
#> post-warmup draws per chain=5000, total post-warmup draws=20000.
#>
#>        mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
#> beta  -1.12    0.01 0.61 -2.44 -1.51 -1.06 -0.68 -0.09  3349    1
#> alpha  5.17    0.05 2.94  0.14  3.05  4.95  7.03 11.61  3280    1
#> lp__  -3.49    0.02 1.06 -6.39 -3.92 -3.17 -2.73 -2.44  4904    1
#>
#> Samples were drawn using NUTS(diag_e) at Thu Mar 18 20:50:28 2021.
#> For each parameter, n_eff is a crude measure of effective sample size,
#> and Rhat is the potential scale reduction factor on split chains (at
#> convergence, Rhat=1).
```