Bayesian Logistic Regression

Ayush Agarwal

Date Submitted : Monday, 04/11/2019

Premise

For $i = 1, ..., n$; Let $x_i = (1, x_{i2}, ..., x_{i5})^T$ be the vector of covariates for the i_{th} observation and $\beta \in \mathbb{R}^5$ be the corresponding vector of regression coefficients.

Suppose response y_i is a realization of Y_i with :

$Y_i \sim \mathsf{Bern}(p_i)$; $p_i = \frac{exp(x_i^T \beta)}{1+exp(x_i^T \beta)}$

Since this is a Bayesian model, we also assume that β has the following prior distribution :

$\beta \sim \mathsf{N}_5(0, 100I_5)$

Objective

Our goal is to use a Bayesian Model to find out the **posterior distribution** for the regression cofficient vector β using Markov Chain Monte Carlo (MCMC) simulation. We will report the posterior mean $E[\beta|y]$ as our final estimate.

Theoritical Background

We will assume a prior distribution on the parameter that we will feed to the model, denoted as $\mathsf{\Pi}(p)$ where p is the parameter.

Now we will take into account our observed data,

$Y_1, Y_2, ...Y_n \sim F(p)$

and *update* the prior distribution to agree with the said data and obtain a **posterior distribution** denoted by $\mathsf{\Pi}(p|\tilde{y})$,

Here $\tilde{y} = (Y_1, Y_2, ...Y_n)$

Equation for Posterior Distribution

$\mathsf{\Pi}(p|\tilde{y})$ α $\mathsf{f}(\tilde{y}|p) = \mathsf{f}(\tilde{y}, p) * \mathsf{\Pi}(p)$, (Using Bayesian Rule)

Where $\mathsf{f}(\tilde{y}|p)$ is the joint likelihood of the data given parameter.

Now, we have a target distribution $\mathsf{\Pi}(p|\tilde{y})$ or $\tilde{\mathsf{\Pi}}(\mathsf{x})$.

We sample $x_1, x_2, ..., x_n \sim \tilde{\mathsf{\Pi}}(\mathsf{x})$ and use the sample statistics as estimators.

Markov Chain Monte Carlo

We will sample a Markov Chain from a stationary distribution $\tilde{\Pi}(x)$. Let $\tilde{\Pi}(x)$ be defined on the space χ. We choose a starting value $x_1 \sim \tilde{\Pi}(x)$. A *Markov Transition Kernel* tells us the next value

Let $A \subseteq \chi$

$P(x_1, A) := Pr(x_2 \in A | X_1 = x_1)$ (Probability of jumping to A, given X_1) Note, $P(x_1, \chi) = 1$

$\tilde{\Pi}(x)$ is called a *stationary distribution* if :-

$P(x_1, x_2) * \tilde{\Pi}(x_1) = Pr(X_2 = x_2, X_1 = x_1)$

OR,

$\int P(x_1, x_2)\tilde{\Pi}(x_1)dx_1 = \tilde{\Pi}(x_2)$

In practice, since it is difficult to draw even $x_1 \sim \tilde{\Pi}(x)$. If we choose $x_1 \in \chi$ and run the Markov Chain, then, $x_\infty \sim \tilde{\Pi}(x)$

For reasonably large T, we get $x_1, x_2, ..., x_T \overset{approx}{\sim} \tilde{\Pi}(x)$

Note, these samples are correlated.

Metropolis-Hastings Algorithm

We want to construct a Markov Chain X_t with stationary distribution Π

Choose a proposal distribution : $q(y|x)$

1. Choose $x_1 \in \chi$, For any t:
2. Draw a proposal $y^* \sim q(y^*|x_t)$
3. Calculate the MH ratio : $\alpha(x_t, y^*) = min(1, \frac{\Pi(y^*)q(x_t|y^*)}{\Pi(x_t)q(y^*|x_t)})$
4. Draw $U \sim Unif[0, 1]$
5. If $U < \alpha(x_t, y^*)$, set $x_{t+1} = y^*$
6. Else, $x_{t+1} = x_t$
7. Stop at $t = T$

(For a symmetric $q(y|x)$, the q-ratio cancels out)

Algorithm for the given Problem

- Using the prior $\Pi(p)$ and the data distribution $F(p)$, obtain the posterior distribution $\tilde{\Pi}(p)$
- Choose appropriate starting value. For this value, using a suitable proposal distribution $q(y|x)$, and step size h, run the MH Algorithm to obtain the MCMC chain.
- Tune the step size h, to get the acceptance probability between 0.23 - 0.30
- Using LLN, we conculde $\frac{\sum_{i=1}^{T} x_i}{T} = \hat{x}$, which is the required estimate.

Computations

Calculating Posterior Distribution

Given Prior: $\Pi(\beta) = \mathsf{N}_5(0, 100I_5)$ i.e $\beta \sim \mathsf{N}_5(0, 100I_5)$

pdf: $f_{\mathbf{X}}(\beta) = \dfrac{\exp\left(-\frac{1}{2*100}(\beta)^{\mathrm{T}}(\beta)\right)}{\sqrt{(2\pi)^5|\ \mathbf{100}|}}$

Given Data: $Y_i \sim Bern(p_i)$; $p_i = \frac{exp(x_i^T\beta)}{1+exp(x_i^T\beta)}$

pdf: $f(y_i; p_i) = p_i^{y_i}(1-p_i)^{1-y_i}$ for $y_i \in \{0, 1\}$

$f(y_i; p_i|\beta) = (\dfrac{exp(x_i^T\beta)}{1+exp(x_i^T\beta)})^{y_i}(\dfrac{1}{1+exp(x_i^T\beta)})^{1-y_i}$ for $y_i \in \{0, 1\}$

$f(y_i; p_i|\beta) = \dfrac{exp(x_i^T\beta))^{y_i}}{1+exp(x_i^T\beta)}$ for $y_i \in \{0, 1\}$

Let $\tilde{y} = (Y_1, Y_2, ...Y_n)$

Joint likelihood: $\mathsf{f}(\tilde{y}|\beta) = \prod_{i=1}^n f(y_i; p_i|\beta) = \prod_{i=1}^n \frac{exp(x_i^T\beta))^{y_i}}{1+exp(x_i^T\beta)}$

Posterior distribution:-

$\Pi(p|\tilde{y}) = \alpha\ \mathsf{f}(\tilde{y}|p) = \mathsf{f}(\tilde{y}, p) * \Pi(p)$

$\Pi(p|\tilde{y}) = \prod_{i=1}^n \frac{exp(x_i^T\beta))^{y_i}}{1+exp(x_i^T\beta)} \exp(-\frac{1}{2*100}\beta^T\beta)$

Log Posterior distribution:-

$\log(\Pi(p|\tilde{y})) = \log(\prod_{i=1}^n \frac{exp(x_i^T\beta))^{y_i}}{1+exp(x_i^T\beta)}) \log(\exp(-\frac{1}{2*100}\beta^T\beta))$

$\log(\Pi(p|\tilde{y})) = (-\frac{1}{2*100}\beta^T\beta) + \sum_{i=1}^n((x_i^T\beta))y_i) - \log(\sum_{i=1}^n(1 + exp(x_i^T\beta)))$

Choosing a Proposal distribution & a Starting value

Since here, $\chi = \mathbb{R}^5$, For X_{t+1} we can use a Normal Proposal centred around X_t, with appropriate step size h as the covariance matrix.

i.e $q(x_{t+1}|x_t) \sim N_5(x_t, h)$

```
Note: Since q is symmetric, MH ratio simplifies.
```

Now, for a starting value we may choose to look at an ML estimate given the data, but ML estimate is unlikely to have a closed form. So, to resolve the issue we will simulate 10,000 values of β from its prior distribution and calculate the log likelihood for each of these values, we pick β_0 as

$\beta_0 = \underset{\beta}{argmax}(f(\tilde{y}|\beta)$

R Code

```r
set.seed(42)

logpost <- function(y,x,beta,sig2,flag=0) #calculate posterior likelihood (log scale)
{
  n = length(y)
```

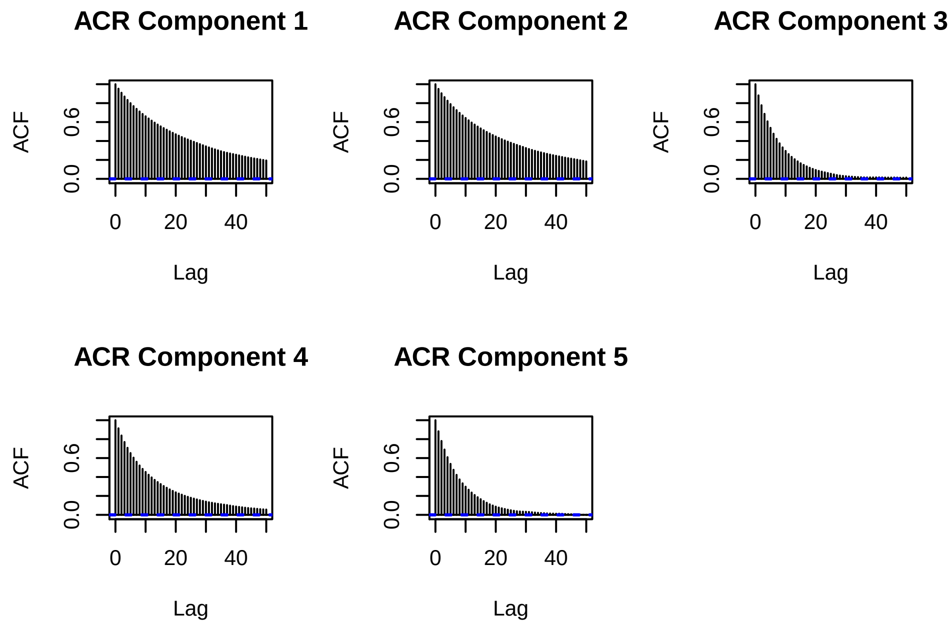
```r
  like  = 0

  for(i in 1:n)
  {
    like = like + (t(y[i]*x[i,]) %*% (beta)) - log(1 + exp(t(x[i,]) %*% (beta)))
  }

  if(flag==1) #for betastart (initialization)
  {
    return(like)
  }

  else{
        like = like - ((t(beta) %*% (beta))/(2*sig2))
        return(like)
  }
}

RMVNorm <- function(n,mu,sigma) #Draws Randomly from MVNorm
{
  p <- length(mu)
  decomp <- svd(sigma)      #using svd instead of cholesky
  sqrt.sig <- decomp$u %*% diag(sqrt(decomp$d), p) %*% t(decomp$v)

  sample <- matrix(0,nrow=n,ncol=p)
  for(i in 1:n)
  {
    Z <- rnorm(p, mean = 0, sd = 1)
    sample[i,] = mu + sqrt.sig %*% Z
  }
  return(sample)
}

##  Finding initialising value by sampling from Prior and  ##
##  Using the Beta = argmax(likelihood(Beta)) for the data ##

FindBetaStart <- function(beta,y,x)
{
  likelihood <- numeric(length = nrow(beta))

  for(i in 1:nrow(beta))
  {
    likelihood[i] <- logpost(y=y,x=x,beta=as.matrix(beta[i,]),100,flag=1)
  }

  beta.init <- beta[which.max(likelihood),]
  return(beta.init)
}

BETAmcmc <- function(y,x,h,N,start,acc.prob) #MCMC Sampling
{
  chain = matrix(0,nrow = N,ncol = length(start))
  chain[1,] = start
```

```
  naccept <- 0

  for(t in 2:N)
  {
      prop <- as.vector(RMVNorm(n=1,mu=chain[t-1,],sigma=h))
      ratio <- logpost(y=y,x=x,beta=prop,sig2=100,flag=0) - logpost(y=y,x=x,beta=as.vector(chain[t-1,])
      if(runif(1,min=0,max=1) < exp(ratio))
      {
        naccept <- naccept + 1
        chain[t,] <- prop
      }

      else
      {
        chain[t,] <- chain[t-1,]
      }
  }

  acc.prob <<- naccept/N              #Global variable assignment
  return(chain)
}


dat <- read.table("http://home.iitk.ac.in/~dootika/assets/course/Log_data/170187.txt",
                  header = F)
Ydat <- as.matrix(dat[,1])
Xdat <- as.matrix(dat[,-1])
N <- 1e5

#Sample from Prior
beta.sample <- RMVNorm(n=1e4,mu=rep(0,ncol(Xdat)),sigma = 100*diag(ncol(Xdat)))
init <- FindBetaStart(y=Ydat,x=Xdat,beta = beta.sample)

step = 0.155*diag(5) #Update step size
acc.prob = 0
chain <- BETAmcmc(y=Ydat,x=Xdat,h=step,N=N,start = init,acc.prob=acc.prob)

beta.est <- colMeans(chain) #Posterior Mean
```

Relevant Plots

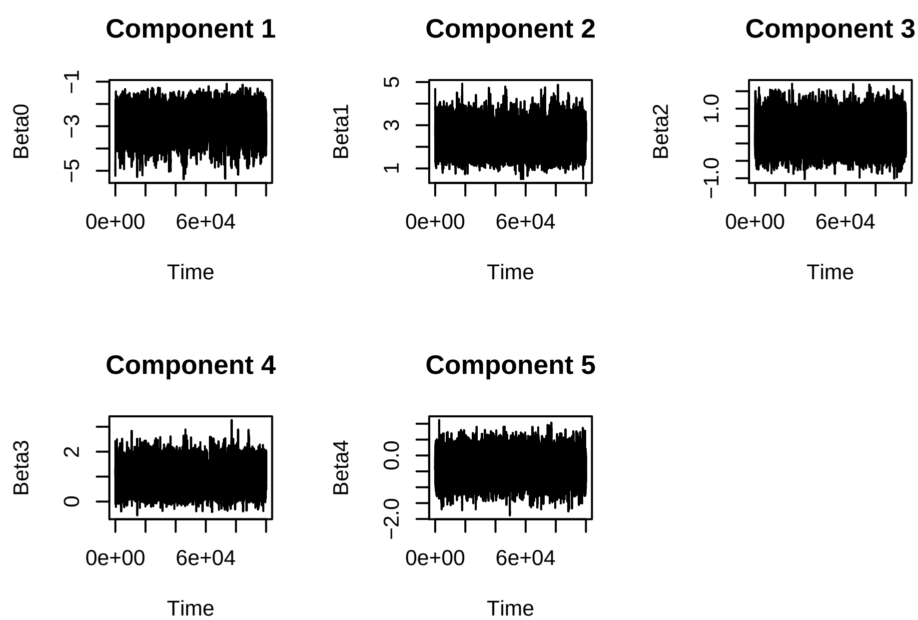
Auto Correlation Plots

```
par(mfrow=c(2,3)) # Auto Correlation Plots
acf(chain[,1],main="ACR Component 1")
acf(chain[,2],main="ACR Component 2")
acf(chain[,3],main="ACR Component 3")
acf(chain[,4],main="ACR Component 4")
acf(chain[,5],main="ACR Component 5")
```



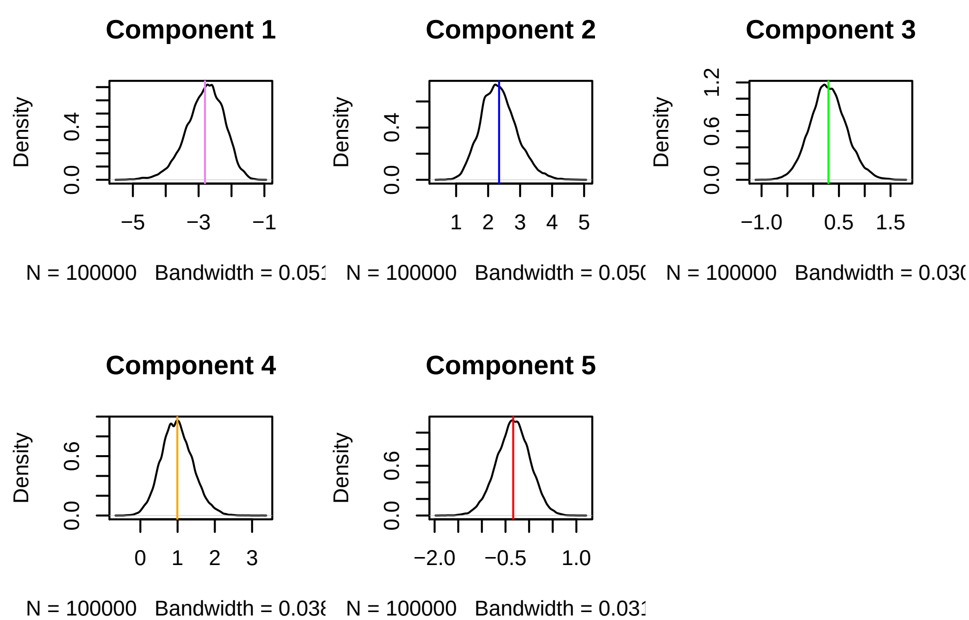
Time Plots

```
par(mfrow=c(2,3)) #Timed Plots
plot.ts(chain[,1],main="Component 1",ylab="Beta0")
plot.ts(chain[,2],main="Component 2",ylab="Beta1")
plot.ts(chain[,3],main="Component 3",ylab="Beta2")
plot.ts(chain[,4],main="Component 4",ylab="Beta3")
plot.ts(chain[,5],main="Component 5",ylab="Beta4")
```



Density Plots

```r
par(mfrow=c(2,3)) #Density Plots
plot(density(chain[,1]),main="Component 1")
abline(v=mean(chain[,1]),col="violet")
plot(density(chain[,2]),main="Component 2")
abline(v=mean(chain[,2]),col="blue")
plot(density(chain[,3]),main="Component 3")
abline(v=mean(chain[,3]),col="green")
plot(density(chain[,4]),main="Component 4")
abline(v=mean(chain[,4]),col="orange")
plot(density(chain[,5]),main="Component 5")
abline(v=mean(chain[,5]),col="red")
```



Estimate and Accpetance probability

```r
print(acc.prob)
```

```
## [1] 0.2765
```

```r
print(beta.est)
```

```
## [1] -2.8068804  2.3423741  0.2961535  0.9920754 -0.3368211
```