Logistic regression

Mats Hansson

16 april 2016

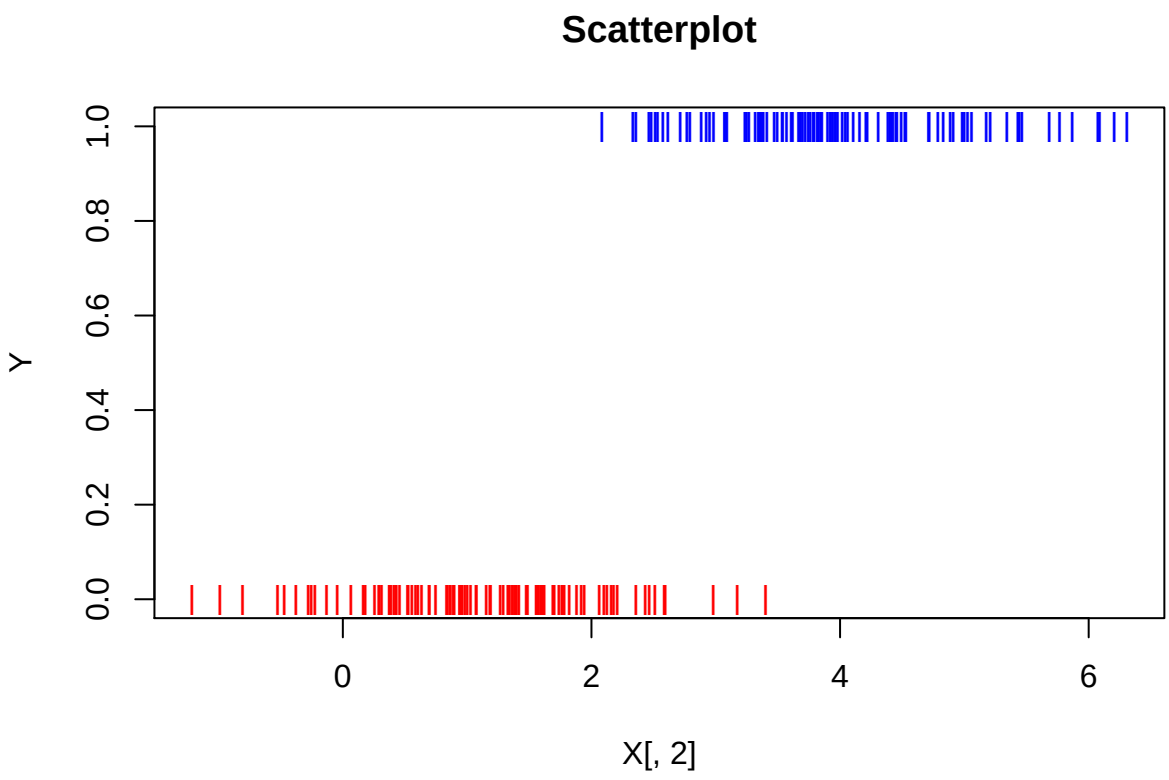
How to bulid a logistic regression

First we have to create some data, the simulated data is normal distributed with a mean at 1 and a variance is 1. The other simulated data is also normal distributed with a mean at 4 and the variance is 1.

```
#Predictor variables
set.seed(1)
X <- cbind(1,c(rnorm(100, 1, 1), rnorm(100,4,1)))


#Response variable
Y <- matrix(data=c(rep(0,100), rep(1,100)))
```

Plot the data

```
color=c(rep("red", 100),rep("blue", 100))
plot(X[,2],Y, pch='|', col=color, main = "Scatterplot")
```



Before, we start with actual cost function. Recall the logistic regression hypothesis is defined as:

$$h_\theta(x) = \theta^T x.$$

Where function g is the sigmoid function. The sigmoid function is defined as:

$$g(y) = \frac{1}{1 + e^{-z}}.$$

Our first step is to implement sigmoid function.

```
#Sigmoid function
sigmoid <- function(z)
{
  g <- 1/(1+exp(-z))
  return(g)
}
```

Now we will implement cost function.

```
#Cost Function
cost <- function(theta)
{
  m <- nrow(X)
  g <- sigmoid(X%*%theta)
  J <- (1/m)*sum((-Y*log(g)) - ((1-Y)*log(1-g)))
  return(J)
}
```

Let's test this cost function with initial theta parameters. We will set theta parameters equal to zero initially and check the cost.

```
#Intial theta
initial_theta <- rep(0,ncol(X))

# initalize theta vector
theta<- c(0,0)

# Number of the observations
m <- nrow(X)


#Cost at inital theta
cost(initial_theta)
```

```
## [1] 0.6931472
```

You will find cost is 0.693 with initial parameters. Now, our objective is to minimize this cost and derive the optimal value of the thetas.

Here I will use inbuilt function of R `optim()` to derive the best fitting parameters. Ultimately we want to have optimal value of the cost function and theta.

```
# Derive theta using gradient descent using optim function
theta_optim <- optim(par=initial_theta,fn=cost)
```

The cofficeint in the model is:

```
#set beta
coef <- theta_optim$par
print(coef)
```

```
## [1] -10.253795   4.066617
```

```
#cost at optimal value of the theta
theta_optim$value
```

```
## [1] 0.1260329
```

We have optimal values of the theta and cost is about 0.1595467 at optimal value of theta. The coefficient is the the odds ratio, therefore the inverse value is probability.

We know that the red observation chould have a mean at 1 and the blue at 4 therefore the best split is (4-1)/2+1 =2.5

the simulated mean is for the red observation is

```
mean(X[1:100,2])
```

```
## [1] 1.108887
```

and for the blue is

```
mean(X[101:200,2])
```

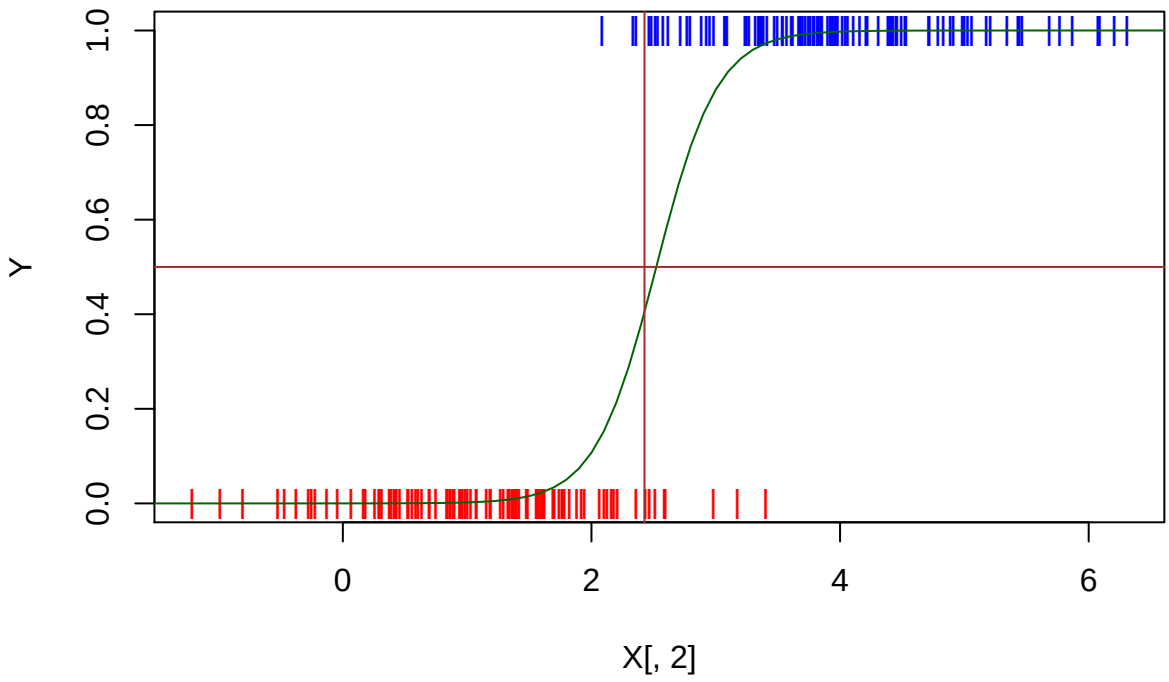
```
## [1] 3.962192
```

therefore the simultated split should be (4.037597- 1.08535)/2+1= 2.476123

If I use the probability, than we can plot the probability curve:

```
### plot ###
color=c(rep("red", 100),rep("blue", 100))
plot(X[,2],Y, pch='|', col=color)

### prediction  ###
xmat=cbind(1, seq(-3, 7, by=0.1))

### the  odds prediction ###
pred_odds<-xmat%*%coef

### The prediction ###
pred=exp(pred_odds)/(1+exp(pred_odds))
```

```
### The prediction line ###
lines(xmat[,2], pred, col="darkgreen")

## the vertical line is
abline(v=(mean(X[101:200,2])-mean(X[1:100,2]))/2+1, col="brown")

## The horizontial line is
abline(h=0.5, col="brown")
```



and if the prediction line is correct should all the observation that is less than 0.5 on the Y-axel classify as 0, otherwise 1. We can also see that simultated split is 2.5 on the x-axle, where it cross the prediction line at 0.5 on the Y-axle.