CUNY School of Professional Studies

SPS.CUNY.EDU

Lecture 06
2020 Spring Data-622
Logistic Regression
Raman Kannan

Instructor Email Address: Raman.Kannan@sps.cuny.edu

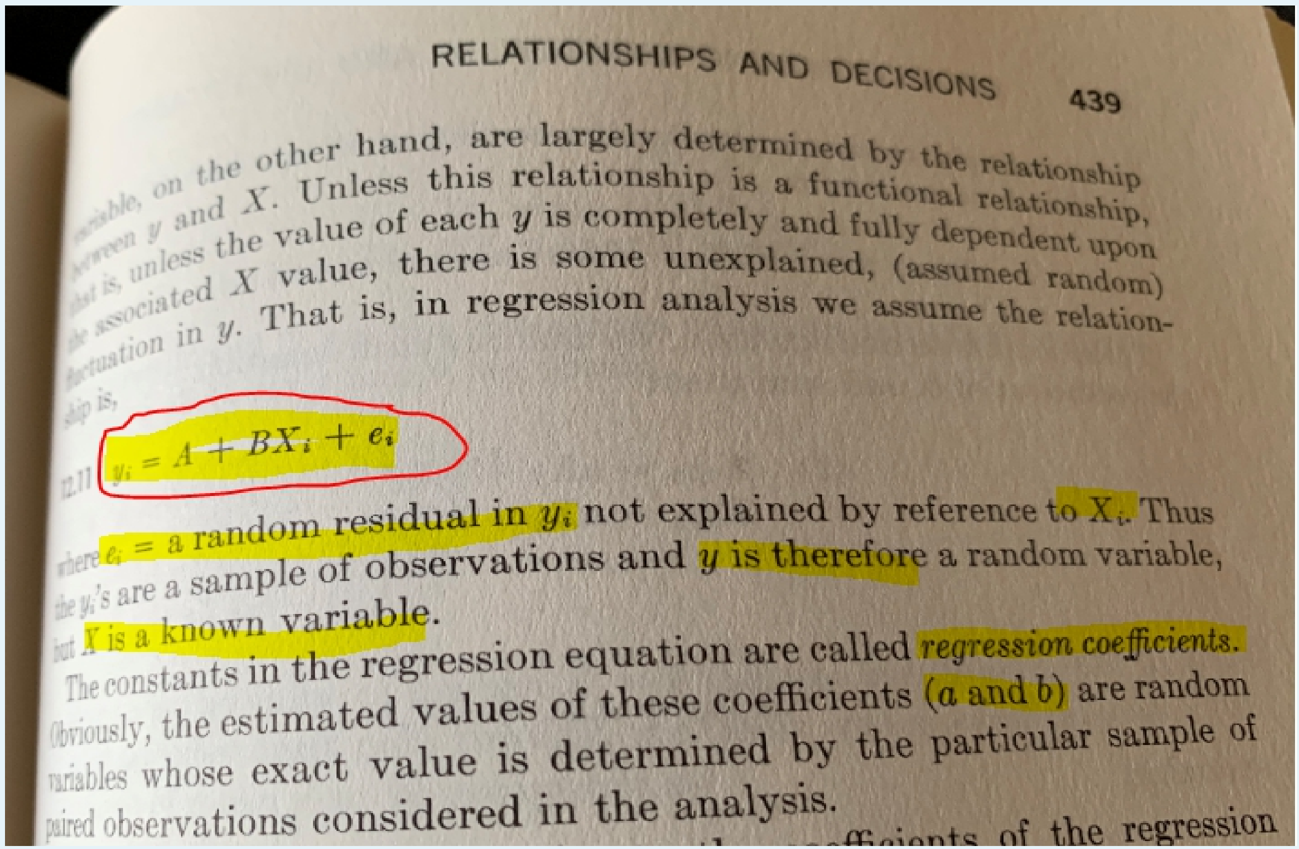
Acknowledgements:
Generous support from IBM Power Systems Academic Initiative
IBM PSAI provides computing infrastructure for free

Script for Algorithms

Develop the Intuition
Understand the assumptions
Develop the mathematics
Run the algorithms
Learn to interpret the result/output
Predict using the model
Learn to determine the performance
Distinguish training/testing error
Differentiate between overfitting/underfitting
Techniques to improve performance

OLS Model

Introduction to Quantitative Management – George J. Brabb



RELATIONSHIPS AND DECISIONS 439

variable, on the other hand, are largely determined by the relationship between y and X. Unless this relationship is a functional relationship, that is, unless the value of each y is completely and fully dependent upon the associated X value, there is some unexplained, (assumed random) fluctuation in y. That is, in regression analysis we assume the relationship is,

12.11 $$y_i = A + BX_i + e_i$$

where e_i = a random residual in y_i not explained by reference to X_i. Thus the y_i's are a sample of observations and y is therefore a random variable, but X is a known variable.

The constants in the regression equation are called *regression coefficients.* Obviously, the estimated values of these coefficients (a and b) are random variables whose exact value is determined by the particular sample of paired observations considered in the analysis.

Need for a Logistic Function

Recall, $y = Xb + e$

In some scenarios, y, the DV that we wish to compute using observable Xs is not continuous. It is dichothomous 1 or 0.

Consider, medical diagnostics, given some test results (X), physician has to determine if the patient is pregnant (YES or NO) or malignant or benign cancer.
Or Admissions committee, has to decide whether to admit or reject a student
Or a bank has to extend or decline credit to a customer.

There are so many instances where the dependent variable is dichotomous – can take one of two values. Binary Classification. When there are more than two, it is called Multi Class Classification.

Logistic Function

Let's understand how Logistic Regression works. For Linear Regression, where the output is a linear combination of input feature(s), we write the equation as:

```
`Y = βo + β1X + ∈`
```

In Logistic Regression, we use the same equation but with some modifications made to `Y`. Let's reiterate a fact about Logistic Regression: we calculate probabilities. And, probabilities always lie between 0 and 1. In other words, we can say:

1. The response value must be positive.
2. It should be lower than 1.

First, we'll meet the above two criteria. We know the exponential of any value is always a positive number. And, any `number` divided by `number + 1` will always be lower than 1. Let's implement these two findings:

$$P(Y=1|X) = \frac{e^{(\beta_o+\beta_1 x)}}{e^{(\beta_o+\beta_1 x)}+1}$$

This is the logistic function.

https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/logistic-regression-analysis-r/tutorial/

As a function of x

https://www.hackerearth.com/blog/wp-content/uploads/2017/01/equateimage.png

NY EMAIL ISL Statistics How to Stop Skype Fr Attributes: Defining C Fundry & Yesco - Firm

$$\implies p(X) = \frac{e^{(\beta_o + \beta_1 x)}}{e^{(\beta_o + \beta_1 x)} + 1}$$

$$\implies p(e^{(\beta_o + \beta_1 x)} + 1) = e^{(\beta_o + \beta_1 x)}$$

$$\implies p.e^{(\beta_o + \beta_1 x)} + p = e^{(\beta_o + \beta_1 x)}$$

$$\implies p = e^{(\beta_o + \beta_1 x)} - p.e^{(\beta_o + \beta_1 x)}$$

$$\implies p = e^{(\beta_o + \beta_1 x)}(1 - p)$$

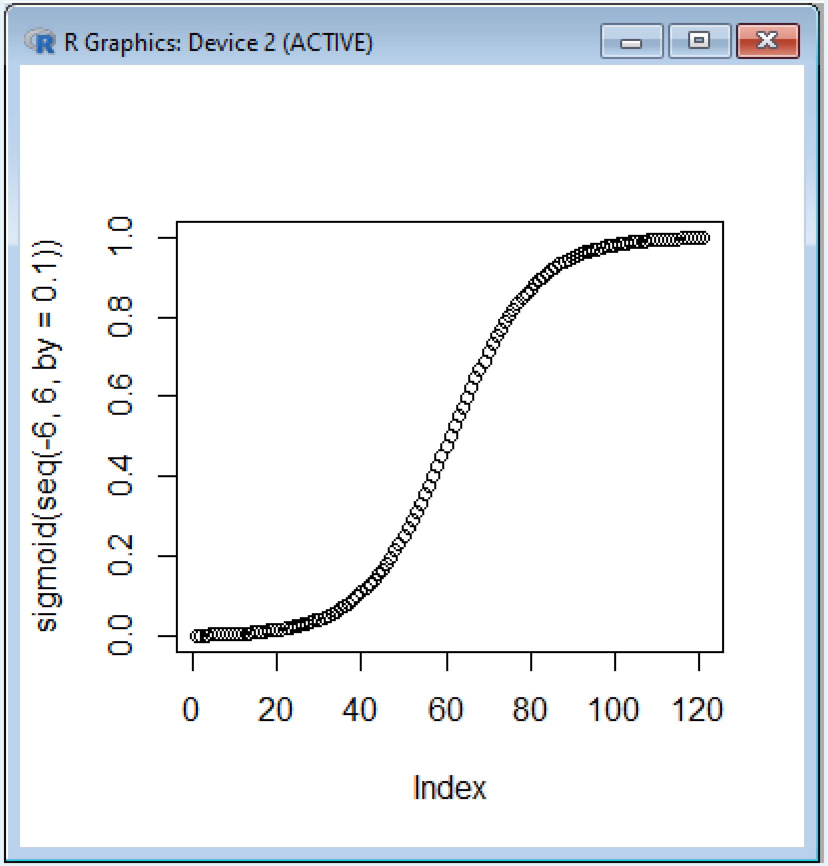
$$\implies \frac{p}{1-p} = e^{(\beta_o + \beta_1 x)}$$

$$\implies \ln(\tfrac{p}{1-p}) = \beta_0 + \beta_1 x$$

In search of a Transformation

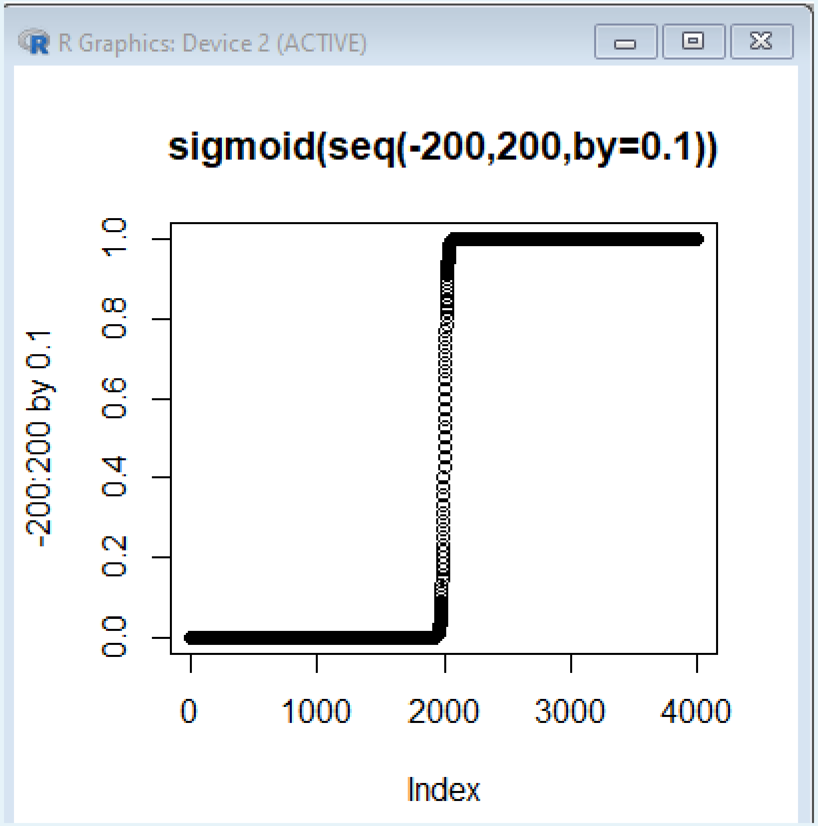
For binary classification, when the allowed values are either 0 or 1 we need a function that transforms X into one of two values.
One such function is the sigmoid function

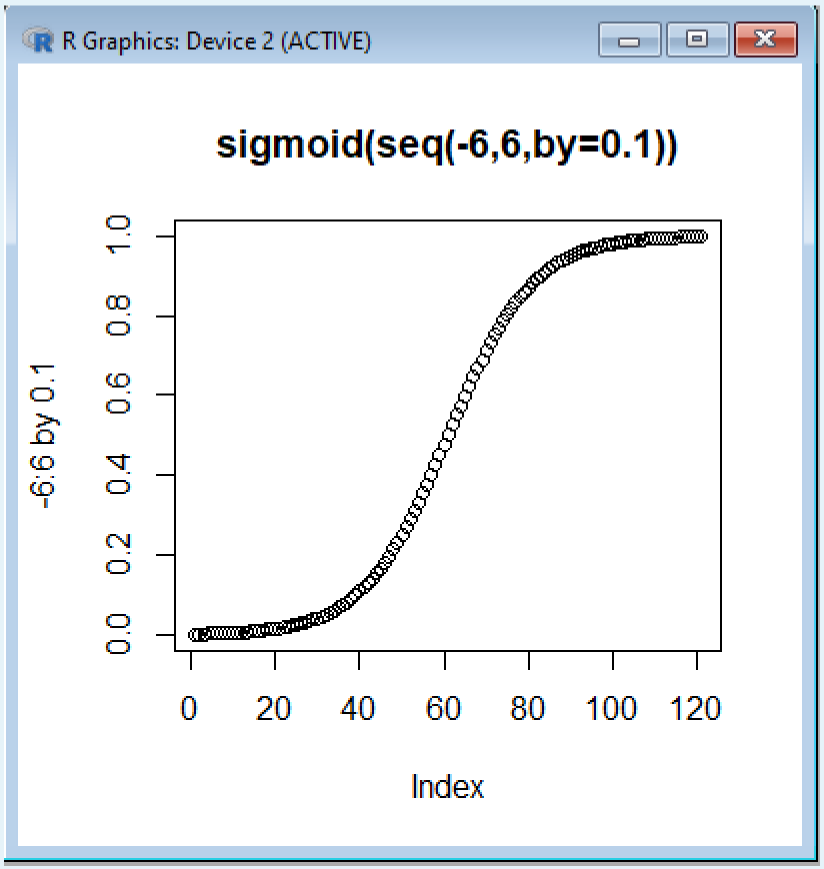
```
vecp<-seq(-6,6,by=0.1)
sigmoid <- function(x){ 1/(1+exp(-x))}
plot(sigmoid(seq(-6,6,by=0.1)) )
```



In search of a Transformation

plot(sigmoid(seq(-200,200,by=0.1)),ylab="-200:200 by 0.1");title("sigmoid(seq(-200,200,by=0.1))")





plot(sigmoid(seq(-6,6,by=0.1)),ylab="-6:6 by 0.1");title("sigmoid(seq(-6,6,by=0.1))")

Logistic H: *sigmoid <- function(x){ 1/(1+exp(-x))}*

Where $X=\Sigma\beta_i{*}X_i$

glm, the generalize linear model function in R computes the beta

```
logisticdata<-read.csv("https://pingax.com/wp-content/uploads/2013/12/data.csv")
plot(logisticdata$score.1,logisticdata$score.2,col=as.factor(logisticdata$label),
xlab="Score-1",ylab="Score-2") # view the data
```

Let us prepare data for training and testing

```
ccases1<-na.omit(logisticdata) #OR complete.cases , keep only complete observations
ccases2<-logisticdata[complete.cases(logisticdata),]
nrow(ccases1)
allidx<-1:nrow(ccases1)
set.seed(1313) # we are sampling for repeatability we set the seed
trainidx<-sample(allidx,round(0.7*nrow(ccases1)),replace=F)
traindata<-ccases1[trainidx,]
testdata<-ccases1[-trainidx,]
table(traindata$label)# check to make sure test and train are similarly distributed
table(testdata$label)
```

Running the model

we will run glm to generate
the model (betas) on
training data

glm.model<-
glm(label~.,
data=traindata,
family='binomial')

summary(glm.model)

coef(glm.model)

The null deviance is 3 times the deviance with the model. So model is improving..

P-values are all small and we can reject the NULL

The model is significant.

```
>
> glm.model <-glm.model<-glm(label~.,data=traindata,family='binomial')
>
> summary(glm.model)

Call:
glm(formula = label ~ ., family = "binomial", data = traindata)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.06587  -0.31121   0.02828   0.27766   1.82200

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -22.70236    6.29030  -3.609 0.000307 ***
score.1       0.18714    0.05128   3.649 0.000263 ***
score.2       0.17713    0.05237   3.382 0.000718 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 92.360  on 69  degrees of freedom
Residual deviance: 35.151  on 67  degrees of freedom
AIC: 41.151

Number of Fisher Scoring iterations: 7

>
> coef(glm.model)
(Intercept)     score.1     score.2
-22.7023627   0.1871382   0.1771322
>
```

Predicting using the model coeff

```
> computed.train.probabilities<-sigmoid(as.matrix(data.frame(intercept=1,traindata$score.1,traindata$score.2)) %*% betavec)
> computed.test.probabilities<-sigmoid(as.matrix(data.frame(intercept=1,testdata$score.1,testdata$score.2)) %*% betavec)
> computed.test.labels<-ifelse(computed.test.probabilities>0.5,1,0)
> computed.train.labels<-ifelse(computed.train.probabilities>0.5,1,0)
> table(computed.train.labels==traindata$label)

FALSE  TRUE 
    9    61 
> table(computed.test.labels==testdata$label)

FALSE  TRUE 
    1    29 
```

```
 computed.train.probabilities<-sigmoid(as.matrix(
data.frame(intercept=1,traindata$score.1,traindata$score.2)) %*% betavec)
 computed.test.probabilities<-sigmoid(as.matrix(
data.frame(intercept=1,testdata$score.1,testdata$score.2)) %*% betavec)
 computed.test.labels<-ifelse(computed.test.probabilities>0.5,1,0)
 computed.train.labels<-ifelse(computed.train.probabilities>0.5,1,0)
 table(computed.train.labels==traindata$label)
 table(computed.test.labels==testdata$label)

plot(1:70,computed.train.probabilities,col=ifelse(computed.train.probabilities>0.50
,'black','yellow'),xlab="obs",ylab="probabilities")
>
plot(1:30,computed.test.probabilities,col=ifelse(computed.test.probabilities>0.50,'
black','yellow'),xlab="obs",ylab="probabilities")
```

Predicting using predict

```
> estimated.train.probabilities<-predict(glm.model,newdata=traindata[,c("score.1","score.2")],type='response')
> head(estimated.train.probabilities)
          89           39           70           48           62           61 
0.9998919382 0.2071082482 0.6438325823 0.9999825600 0.0004219775 0.9986386035 
> estimated.train.labels<-ifelse(estimated.train.probabilities>0.5,1,0)
> estimated.test.probabilities<-predict(glm.model,newdata=testdata[,c("score.1","score.2")],type='response')
> estimated.test.labels<-ifelse(estimated.test.probabilities>0.5,1,0)
> table(estimated.test.labels==computed.test.labels)

TRUE 
  30 
> table(estimated.train.labels==computed.train.labels)

TRUE 
  70 
> table(estimated.train.labels==traindata.labels)
Error in table(estimated.train.labels == traindata.labels) : 
  object 'traindata.labels' not found
> table(estimated.train.labels==traindata.label)
Error in table(estimated.train.labels == traindata.label) : 
  object 'traindata.label' not found
> table(estimated.train.labels==traindata$label)

FALSE  TRUE 
    9    61 
> table(estimated.test.labels==testdata$label)

FALSE  TRUE 
    1    29 
```

Predicting using predict

```
estimated.train.probabilities<-
predict(glm.model,newdata=traindata[,c("score.1","score.2")],type='response')
head(estimated.train.probabilities)
estimated.train.labels<-ifelse(estimated.train.probabilities>0.5,1,0)
estimated.test.probabilities<-
predict(glm.model,newdata=testdata[,c("score.1","score.2")],type='response')
estimated.test.labels<-ifelse(estimated.test.probabilities>0.5,1,0)
table(estimated.test.labels==computed.test.labels)
table(estimated.train.labels==computed.train.labels)
table(estimated.train.labels==traindata$label)
table(estimated.test.labels==testdata$label)
```

Performance

```
require(ROCR)
glm_prediction<-prediction(estimated.test.probabilities,testdata$label)
glm_perf<-performance(glm_prediction,measure="tpr",x.measure="fpr")
glm_slot_fp<-slot(glm_prediction,"fp")
glm_slot_tp<-slot(glm_prediction,"tp")
glm_slot_tn<-slot(glm_prediction,"n.neg")
glm_slot_fn<-slot(glm_prediction,"n.pos")
glm_auc<-performance(glm_prediction,"auc")@y.values[[1]]

 plot(unlist(glm_slot_fp)/unlist(glm_slot_tn),
unlist(glm_slot_tp)/unlist(glm_slot_fn),main="ROCR
Curve",xlab="FPR",ylab='TPR')
```

References

https://www.statmethods.net/advstats/glm.htm
https://www.r-bloggers.com/logistic-regression-with-r-step-by-step-implementation-part-2
http://pingax.com/wp-content/uploads/2013/12/data.csv
https://towardsdatascience.com/understanding-logistic-regression-9b02c2aec102
https://www.hackerearth.com/blog/wp-content/uploads/2017/01/equateimage.png

https://www.hackerearth.com/blog/wp-content/uploads/2017/01/equateimage.png