Lab #9 - Logistic Regression Part I

Econ 224

September 25th, 2018

Introduction

In this lab we'll study logistic regression. The first part of the lab will involve carrying out some calculations to better understand how logistic regression works and what it means. The second part of the lab will show you the basics of how to carry out logistic regresion in R.

Part I - Theoretical

In this part of the lab, we'll carry out some theoretical derivations to better understand logistic regression. To make things simpler, we'll use some slightly different notation and terminology than ISL. First we'll define the *column vectors* X and β as follows:

$$X = \begin{bmatrix} 1 \\ X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}$$

Notice that the first element of X is *not* X_1: it is simply the number 1. There's an important reason for this that you'll see in a moment. From the reading, we know that logistic regression is a *linear model* for the *log odds*, namely

$$\log\left[\frac{P(X)}{1-P(X)}\right] = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

where $P(X)$ is shorthand for $\mathbb{P}(Y = 1|X)$. Note that when I write log I **always** mean the natural logarithm. Also note that when I write $\exp(z)$ I mean e^z. This comes in handy if z is a complicated expression.

Using the vector notation introduced above, we can express this more compactly as

$$\log\left[\frac{P(X)}{1-P(X)}\right] = X'\beta$$

since

$$X'\beta = \begin{bmatrix} 1 & X_1 & X_2 & \cdots & X_p \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} = \beta_0 + \beta_1 X_2 + \beta_2 X_2 + \cdots + \beta_p X_p$$

I will call $X'\beta$ the *linear predictor* since it is the linear function of X that we use to predict Y. By exponentiating both sides of the log-odds expression from above and re-arranging, obtain the following:

$$
\begin{aligned}
\frac{P(X)}{1-P(X)} &= \exp(X'\beta) \\
P(X) &= [1-P(X)]\exp(X'\beta) \\
P(X) + P(X)\exp(X'\beta) &= \exp(X'\beta) \\
P(X)[1+\exp(X'\beta)] &= \exp(X'\beta) \\
P(X) &= \frac{\exp(X'\beta)}{1+\exp(X'\beta)} \\
P(X) &= \Lambda(X'\beta)
\end{aligned}
$$

where the function Λ is defined as follows

$$\Lambda(z) = \frac{e^z}{1+e^z}$$

Exercise #1

(a) Verify that $\Lambda(z) = \dfrac{1}{1+e^{-z}}$.
(b) Using (b), write an alternative expression for $P(X)$.

Solution to Exercise #1

(a) Dividing the numerator and denominator by e^z, which cannot result in division by zero since e^z is always positive, we have

$$\Lambda(z) = \frac{e^z}{1+e^z} = \frac{1}{1/e^z+1} = \frac{1}{1+e^{-z}}$$

(b) $P(X) = \dfrac{1}{1+\exp(-X'\beta)}$

Interpreting β in a Logistic Regression

From the expression above, we see that β_j is the partial derivative of the log-odds with respect to X_j. But it's difficult to think in terms of log-odds. By doing some calculus (see the exercises below), we can work out the partial derivative of $p(X)$ with respect to X_j, but this will *not* turn out to equal β_j. Because $P(X)$ is not a linear function of X, the derivative varies with X, which makes things fairly complicated. There are two main approaches for dealing with this problem. One is to evaluate the derivative at a "typical" value of X such as the sample mean. Another is to use the "divide by 4 rule." This rule says that if we increase X_j by one unit, $P(X)$ will change by *no more than* $\beta_j/4$. In the following exercise, you'll derive this rule.

Exercise #2

(a) Analyze the function $\Lambda(z)$: calculate its derivative, and its limits as $z \to -\infty$ and $+\infty$. What values can this function take? Is it increasing? Decreasing? Explain.
(b) Use the chain rule and your answer to (a) to find the partial derivative of $\Lambda(X'\beta)$ with respect to X_j.
(c) What is the maximum value of the *derivative* of $\Lambda(z)$? At what value of z does it occur?
(d) Use your answers to parts (a), (b) and (c) to justify the "divide by 4 rule."
(e) The "divide by 4 rule" provides an upper bound on the effect of X_j on $P(X)$. When is this upper bound close to the derivative you calculated in part (c)?

Solution to Exercise #2

(a) The function Λ takes values between 0 and 1. When $z = 0$, $\Lambda(z) = e^0/(1+e^0) = 1/2$. As $z \to \infty$, $\Lambda(z) \to 1$ and as $z \to -\infty$, $\Lambda(z) \to 0$. We calculate its derivative using the quotient rule as follows

$$\frac{d\Lambda(z)}{dz} = \frac{e^z(1+e^z) - e^z e^z}{(1+e^z)^2} = \frac{e^z}{(1+e^z)^2}$$

Since e^z is always greater than zero, the derivative is always positive so $\Lambda(z)$ is strictly increasing.

(b) The key is to treat the linear predictor $X'\beta$ as a function of X_j, namely

$$f(X_j) = X'\beta = \beta_0 + \beta_1 X_1 + \cdots + \beta_j X_j + \beta_{j+1} X_{j+1} + \cdots + \beta_p X_p$$

Now, by the chain rule we have

$$\frac{\partial \Lambda(X'\beta)}{\partial X_j} = \frac{\partial \Lambda\left(f(X_j)\right)}{\partial X_j} = \frac{\exp(X'\beta)}{[1+\exp(X'\beta)]^2}\frac{\partial f(X_j)}{\partial X_j} = \frac{\beta_j \exp(X'\beta)}{[1+\exp(X'\beta)]^2}$$

(c) To find the value of z that maximizes the first derivative, we take the *second* derivative of Λ as follows

$$\begin{aligned}
\frac{d^2\Lambda(z)}{dz} &= \frac{e^z(1+e^z)^2 - 2e^z(1+e^z)e^z}{(1+e^z)^4} = \frac{e^z(1+2e^z+e^{2z}) - 2e^{2z}(1+e^z)}{(1+e^z)^4} \\
&= \frac{e^z + 2e^{2z} + e^{3z} - 2e^{2z} - 2e^{3z}}{(1+e^z)^4} = \frac{e^z - e^{3z}}{(1+e^z)^4} \\
&= \frac{e^z(1-e^{2z})}{(1+e^z)^4} = \frac{e^z(1+e^z)(1-e^z)}{(1+e^z)^4} = \frac{e^z(1-e^z)}{(1+e^z)^3}
\end{aligned}$$

Thus, the first order condition is $e^z(1-e^z) = 0$. Since e^z cannot equal zero for any z, the only way for this equation to be satisfied is if $e^z = 1$ which occurs precisely when $z = 0$. Substituting into our expression from (a), we find that the derivative of $\Lambda(z)$ at $z = 0$ is $e^0/(1+e^0)^2 = 1/(1+1)^2 = 1/4$.

(d) From part (a), we know that the derivative of $\Lambda(z)$ equals $e^z/(1+e^z)^2$ which is always positive. From part (c) we know that this derivative is *at most* $1/4$. Therefore, the partial derivative of $\Lambda(X'\beta)$ with respect to X_j is *at most* $\beta_j \times 1/4 = \beta_j/4$.

(e) When $X'\beta \approx 0$ it follows that $\exp(X'\beta)/[1+\exp(X'\beta)]^2 \approx 1/4$ so the "divide by four" rule gives a good approximation to the actual derivative.

The Latent Data Formulation of Logistic Regression

$$y_i^* = X_i'\beta + \epsilon_i, \quad y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{if } y_i^* \leq 0 \end{cases}, \quad \epsilon_i \sim \text{ iid Logistic}(0,1)$$

where the Logistic$(0,1)$ distribution has CDF $\Lambda(z) = e^z/(1+e^z)$ and pdf $\lambda(z) = e^z/(1+e^z)^2$. These expressions should look familiar since we worked with them above! Notice that we have

$$\mathbb{P}(y_i = 1) = \mathbb{P}(y_i^* > 0) = \mathbb{P}(X_i'\beta + \epsilon_i > 0) = \mathbb{P}(-\epsilon_i < X_i'\beta) = \mathbb{P}(-\epsilon_i \leq X_i'\beta) = \Lambda(X_i'\beta)$$

where the second-to-last equality follows from the fact that ϵ_i is a continuous random variable, and the final equality follows from the fact that the distribution of ϵ_i is *symmetric*. (You can check this by substituting z and $-z$ into the function λ.)

The idea here is that y_i^* is an "unobserved" aka *latent* variable that determines whether the observed indicator y is positive or negative. In specific examples, we can usually give y_i^* a meaningful interpretation. For example, suppose that $y_i = 1$ if person i voted for Hilary Clinton in the 2016 presidential election and X_i contains demographic information, e.g. income, education, race, sex, and age. The latent variable y_i^* can be viewed as a measure of person i's *strength of preference* for Hilary Clinton relative to Donald Trump. If y_i^* is large and positive, person i strongly prefers Clinton; if y_i^* is large and negative, person i strongly prefers Trump; if $y_i^* = 0$, person i is indifferent.

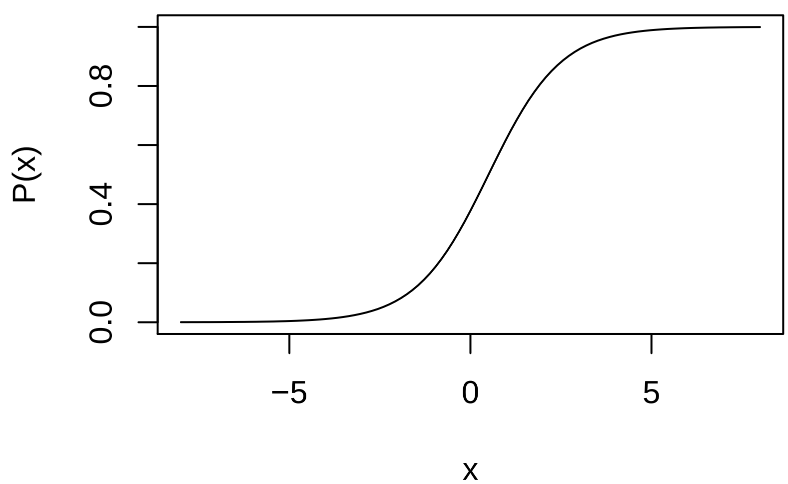
Part II - Running Logistic Regression in R

Now we'll take a quick look at how to carry out logistic regression in R using a simulated dataset. In Thursday's lab you'll use what you learn in this part to study a real-world example.

Generating Data from a Logistic Regression Model

```r
#-------------------------- Simulate data from a logistic regression
library(MASS)
set.seed(1234)
n <- 500
x <- rnorm(n, mean = 1.5, sd = 2)
b0 <- -0.5
b1 <- 1
ystar <- b0 + b1 * x + rlogis(n)
y <- 1 * (ystar > 0)
mydat <- data.frame(x, y)
```

```r
#-------------------------- Plot true logistic regression function for the sim
Lambda <- function(x) {
  1 / (1 + exp(-x))
}
curve(Lambda(b0 + b1 * x), -8, 8, ylab = 'P(x)')
```




```r
#-------------------------- Estimate the coefficients
lreg <- glm(y ~ x, mydat, family = binomial(link = 'logit'))
summary(lreg)
```

```
Call:
glm(formula = y ~ x, family = binomial(link = "logit"), data = mydat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3042  -0.6879   0.2138   0.6570   2.2360
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.78119    0.15389  -5.076 3.85e-07 ***
x            1.07532    0.09972  10.783  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 666.93  on 499  degrees of freedom
Residual deviance: 428.94  on 498  degrees of freedom
AIC: 432.94

Number of Fisher Scoring iterations: 5
```

```r
coef(lreg)
```

```
(Intercept)           x
 -0.7811888   1.0753181
```

```r
#---------------------------- Confidence interval for b0 and b1
-0.78119 + 2 * c(-1, 1) * 0.15389
```

```
[1] -1.08897 -0.47341
```

```r
1.07532 + 2 * c(-1, 1) * 0.09972
```

```
[1] 0.87588 1.27476
```

```r
#--------------------------- Predicted probability that y = 1 when x = 0
# By hand:
bhat_0 <- coef(lreg)[1]
Lambda(bhat_0)
```

```
(Intercept)
  0.3140637
```

```r
# Using predict:
predict(lreg, newdata = data.frame(x = 0), type = 'response')
```

```
        1
0.3140637
```

```r
# What happens if you don't specify type = 'response'?
```

```r
#--------------------------- Predicted probability that y = 1 at the average x
# By hand:
bhat_1 <- coef(lreg)[2]
Lambda(bhat_0 + bhat_1 * mean(x))
```

```
(Intercept)
  0.6975679
```

```r
# Using predict:
predict(lreg, newdata = data.frame(x = mean(x)), type = 'response')
```

```
        1
0.6975679
```

```r
#--------------------------- Marginal effect at average x
linear_predictor <- bhat_0 + bhat_1 * mean(x)
bhat_1 * exp(linear_predictor) / (1 + exp(linear_predictor))^2
```

```
        x
0.2268566
```

```r
# Divide by 4 rule
bhat_1 / 4
```

```
        x
0.2688295
```

```r
#--------------------------- Plot of "Jittered" data and the logistic function
library(ggplot2)
ggplot(mydat, aes(x, y)) +
  geom_jitter(height = 0.1) +
  stat_smooth(method='glm',
              method.args = list(family = "binomial"),
              formula = y ~ x)
```