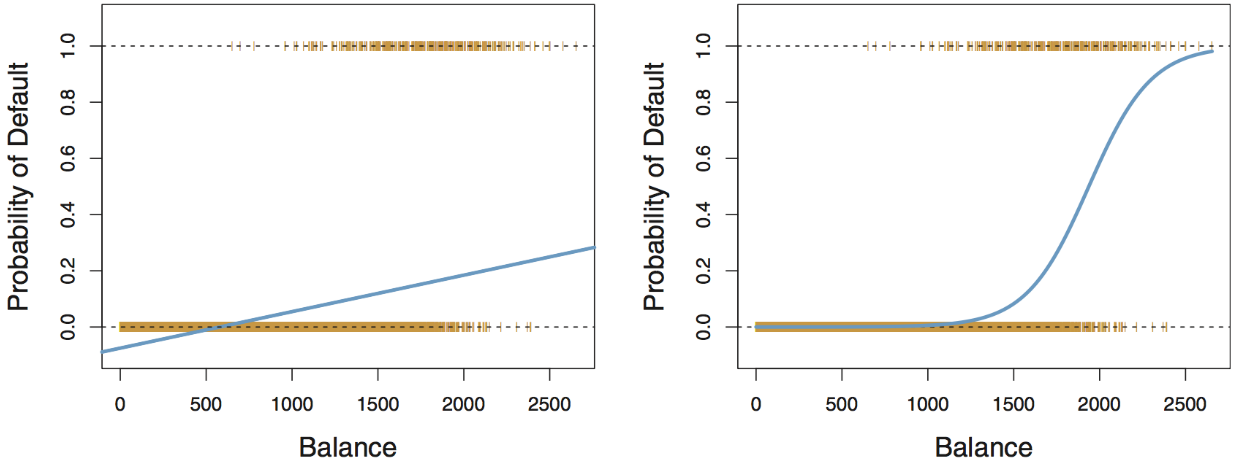
Logistic Regression

36-290 – Statistical Research Methodology

Week 5 Thursday – Fall 2021

The Setting



(Figure 4.2, *Introduction to Statistical Learning* by James et al.)

- To the left is a linear regression fit. The regression line is limited to lay within the range (0,1).
- To the right is a logisitic regression fit. The regression line is limited to lay within the range (0,1).

Logistic Regression

Logistic regression is appropriate for datasets where the response variable can only take on two discrete values (assumed to map to 0 and 1). The underlying distribution is the binomial distribution, whose parameter is p, the probability of success (or of seeing the outcome 1).

For logistic regression, a conventional choice of link function $g(p|\mathbf{x})$ is the *logit* function, which limits the range of the regression line to $(0,1)$:

$$\log\left[\frac{E[Y|\mathbf{x}]}{1-E[Y|\mathbf{x}]}\right] = \log\left[\frac{p}{1-p}\right] = \beta_0 + \sum_{i=1}^{p} \beta_i x_i \,,$$

so that

$$E[Y|\mathbf{x}] = p = \frac{e^{\beta_0+\sum_{i=1}^{p}\beta_i x_i}}{1+e^{\beta_0+\sum_{i=1}^{p}\beta_i x_i}} \,.$$

A major difference between linear and logistic regression is that the latter involves numerical optimization, i.e., instead of plugging into a formula, you have to use an iterative algorithm to find the β's that maximize the likelihood function:

$$\mathcal{L} = \left(\prod_{i:Y_i=1} p_i\right)\left(\prod_{i:Y_i=0} (1-p_i)\right) \,.$$

Numerical optimization means the logistic regression runs more slowly than linear regression.

Logistic Regression: Inference

A major motivating factor underlying the use of logistic regression, and indeed all generalized linear models, is that one can perform inference...e.g., how does the response change when we change a predictor by one unit?

For logistic regression, we utilize the concept of "odds."

Let's say that the predicted response is $p = 0.8$ given a particular predictor variable value. (For simplicity, let's assume we have just one predictor variable.) That means that we expect that if we were to repeatedly sample response values given that predictor variable values, we would expect class 1 to appear four times as often as class 0:

$$O = \frac{E[Y|x]}{1 - E[Y|x]} = \frac{p}{1-p} = \frac{0.8}{1-0.8} = 4 = e^{\beta_0+\beta_1 x} \,.$$

Thus we say that for the given predictor variable value, the odds O are 4 (or 4-1) in favor of class 1.

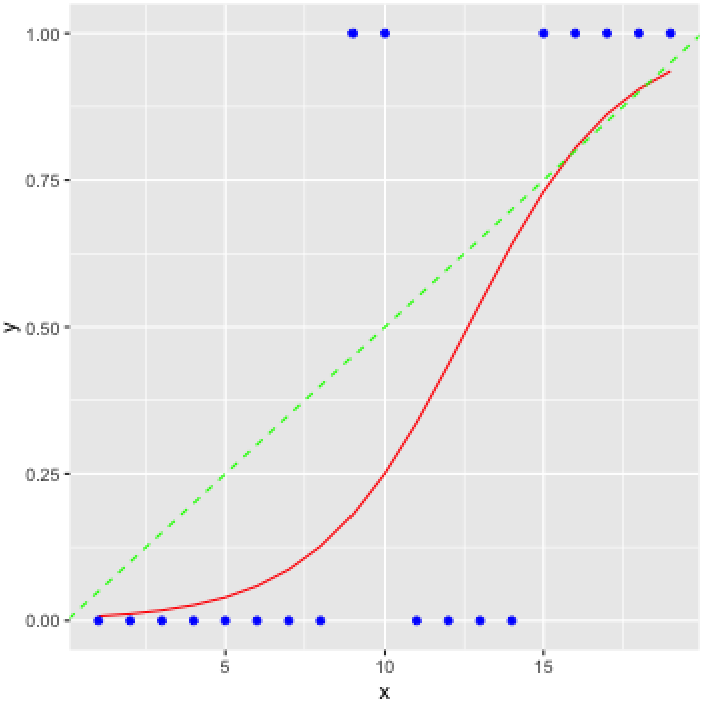
How does the odds change if I change the value of a predictor variable by one unit?

$$O_{\rm new} = e^{\beta_0+\beta_1(x+1)} = e^{\beta_0+\beta_1 x}e^{\beta_1} = e^{\beta_1}O_{\rm old} \,.$$

For every unit change in x, the odds changes by a factor e^{β_1}.

Logistic Regression: Output

```
set.seed(101)
x = 1:19
y = rbinom(length(x),1,0.05*x)
out.log = glm(y~x,family=binomial)
suppressMessages(library(tidyverse))
ggplot(data=data.frame(x=x,y=out.log$fitted.values),mapping=aes(x=x,y=y)) + geom_line(color="red") +
  geom_point(data=data.frame(x=x,y=y),mapping=aes(x=x,y=y),color="blue") +
  geom_abline(slope=0.05,intercept=0,color="green",linetype="dashed")
```



Logistic Regression: Output

```
summary(out.log)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4315  -0.4736  -0.1882   0.4954   1.8504
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.2800     2.3424  -2.254   0.0242 *
## x             0.4186     0.1843   2.271   0.0231 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25.008  on 18  degrees of freedom
## Residual deviance: 14.236  on 17  degrees of freedom
## AIC: 18.236
##
## Number of Fisher Scoring iterations: 5
```

```
logLik(out.log)   # the maximum log-likelihood value
```

```
## 'log Lik.' -7.117803 (df=2)
```

Logistic Regression: Output

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4315  -0.4736  -0.1882   0.4954   1.8504
```

The deviance residuals are, for each datum,

$$d_i = \text{sign}(y_i - \hat{p}_i)\sqrt{-2[y_i \log \hat{p}_i + (1 - y_i)\log(1 - \hat{p}_i)]}\,,$$

where y_i is the i^{th} observed response and $\hat{p}_i$ is the estimated probability of success (i.e., the amplitude of the prediction curve for the i^{th} datum). The sum of squares of the deviance residuals is $-2\log\mathcal{L}$.

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.2800     2.3424  -2.254   0.0242 *
x             0.4186     0.1843   2.271   0.0231 *
```

The intercept of the prediction curve is $e^{-5.28}$ and $O_{\text{new}}/O_{\text{old}} = e^{0.4186}$.

```
    Null deviance: 25.008  on 18  degrees of freedom
Residual deviance: 14.236  on 17  degrees of freedom
AIC: 18.236
...
'log Lik.' -7.117803 (df=2)
```

The maximum value of the log of the likelihood function is -7.118. The residual deviance is -2 times -7.118, or 14.236. The AIC is $2k - 2\log\mathcal{L}$ = $2 \cdot 2 - 2 \cdot (-7.118)$ = 18.236, where k is the number of degrees of freedom (here, `df` = 2). These are all metrics of quality of fit of the model.

Logistic Regression: Predictions

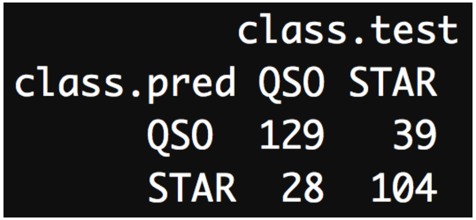
In this example, there was no training/testing split! In "real" analyses, there would be…you'd run on the model and generate test-set predictions via, e.g.,

```
resp.prob = predict(out.log,newdata=pred.test,type="response")
resp.pred = rep(NA,length(resp.prob))
for ( ii in 1:length(resp.prob) ) {
  if (resp.prob[ii] > 0.5) {
    resp.pred[ii] = "<class 1>"    # fill in name of class 1
  } else {
    resp.pred[ii] = "<class 0>"    # fill in name of class 0
  }
}
```

(Note the quotation marks. I'm assuming you are dealing with a factor variable, whose values you refer to by [quoted] name.) `resp.prob` is a number between 0 and 1. If that number is less than 0.5, we predict that the test datum is associated with class 0, otherwise we predict it is associated with class 1. In a future lecture, we will re-examine our use of 0.5 as a threshold for class splitting.

Model Diagnostics: Classification

The most straightforward diagnostic to use to assess logistic regression or any other classification model is the *confusion matrix*, whose rows are predicted classes and whose columns are observed classes:



```
          class.test
class.pred QSO STAR
      QSO  129   39
      STAR  28  104
```

There are *many* metrics associated with confusion matrices. The most basic is the *misclassification rate*, or MCR, which is the ratio of the sum of the off-diagonal values in the confusion matrix (top right and bottom left) to the overall table sum. (For the confusion matrix above, the MCR is 0.223). Other metrics include the *sensitivity* and *specificity*, etc.; for definitions of these and other metrics, see, e.g., this web page.

We will expand upon classification diagnostics in a future lecture.